CombiGlide 2.5

User Manual

Schrödinger Press

June 2009

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, and screen output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: Document Conventions.

In descriptions of command syntax, the following UNIX conventions are used: braces `{ }` enclose a choice of required items, square brackets `[ ]` enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the $ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Introduction to CombiGlide

CombiGlide employs combinatorial technology for lead identification and optimization. It presents three distinct workflows:

1. The Enumerate and Dock workflow—Enumerate a combinatorial library; dock it, analyze and evaluate it the results. This is most useful for relatively small libraries.
2. The Virtual Combinatorial Screening workflow—Use our proprietary technology to explore an extremely large combinatorial space in order to find side chains for a core chemical scaffold that will optimize binding to a receptor of interest.
3. The Core Hopping workflow—Starting with a lead compound in a well-docked pose, find alternative cores that will bind well with a receptor of interest.

All of these can be used either as a prelude to combinatorial design or as a means of conventional lead discovery and optimization. Though CombiGlide carries out combinatorial chemistry in silico, the results may be used either to design focused combinatorial libraries or to evaluate a large universe of compounds for one-off or small-scale automated synthesis, or even for idea generation.

Core hopping is our latest workflow, introduced for the first time in Schrodinger Suite 2007. Just as the Enumerate and Dock workflow and the Virtual Combinatorial Screening workflow hold the core constant and vary the side-chain composition, the Core Hopping workflow holds the side chains of a lead compound constant and varies the core.

This manual contains introductions to all three workflows.

1.1 CombiGlide Features

CombiGlide includes the following capabilities and features, for the three workflows.

Enumerate and Dock:

- Enumerates and docks complete combinatorial libraries on request
- Performs flexible docking using the standard or the extra precision (XP) modes of Glide
- Provides multiple post-docking library selection strategies and options
- Allows for incorporation of predicted ADME properties into selection process

- Analyzes selected libraries for enrichment of actives and chemical features

Virtual combinatorial screening:

- Performs rapid screening of large virtual combinatorial libraries against 3D targets
- Is orders of magnitude faster than docking the entire library
- Performs flexible docking using the standard and extra precision (XP) modes of Glide
- Provides multiple post-docking library selection strategies and options
- Allows for incorporation of predicted ADME properties into selection process
- Analyzes selected libraries for enrichment of actives and chemical features

Core hopping:

- Provides capabilities for substituting the core of a given lead compound

All workflows:

- Provides automated reagent file preparation: 2D to 3D conversion, generation of reasonable ionization and tautomeric states, stereoexpansion, assignment of attachment points
- Uses a "core plus side chains" approach
- Offers extensive flexibility in initial core placement
- Uses an intuitive, user-friendly wizard-based GUI for setting up and monitoring jobs and for visualization of docked poses

1.2 Overview of Core Hopping

In core hopping, the side chains of a lead compound are retained, and the core is substituted by a variety of cores, in the attempt to find a core that might give better binding. The steps in this process are summarized here.

1. Select and prepare the lead compound.

   In this step you define the points at which the side chains are attached to the core. This task is in common with the Virtual Screening Workflow, and is performed in the Define Combinations step of the Combinatorial Screening panel. This step is described in detail in Chapter 7.

2. Prepare receptor and receptor grids.

   The receptor must be prepared for the docking stage, and the receptor grids for the docking must be generated. These tasks are performed in the Protein Preparation Wizard and

Receptor Grid Generation panels, and are described in detail in the *Protein Preparation Guide* and the *Glide User Manual*.

3. Prepare the core scaffold molecules.

   The molecules that contain the alternative cores must be prepared as all-atom, 3D structures, and must be labeled with the attachment points. This step is performed in the Protocore Preparation panel, and is described in Chapter 2. We supply a small "starter" protocore library on the Schrödinger web site.

4. Align and dock the core scaffolds.

   The potential attachment points on the core scaffolds (protocores) are aligned to the side chains of the lead compound (template) during docking to the receptor. The cores are sorted and filtered by goodness of alignment and then redocked into the receptor after attaching the side chains of the template. This step is performed in the Protocore Docking panel, and is described in Chapter 2.

5. Visualize and analyze the results.

   The Core Hopping Visualization panel provides a way to view the docked, dressed core candidates together with the docked lead compound in the active site. It also presents a view of the Project Table that is customized for convenient sorting and selection of promising core structures. This panel is described in Chapter 2.

6. Submit selected cores to the virtual combinatorial screening workflow.

   Though not part of the core hopping workflow per se, selected cores can easily be submitted to the virtual combinatorial screening workflow in order to optimize the side-chain composition.

1.3 Overview of Virtual Combinatorial Screening

CombiGlide can be used for virtual combinatorial screening, in which a focused combinatorial library is obtained at the end of the process. The steps in this process are summarized here. The last five steps constitute the combinatorial screening process (which is summarized with some more detail in Chapter 6).

1. Determine the desired synthetic approach.

   The first step is to identify the chemistry associated with the desired library: the sequence of reactions that leads to the desired product. This step is discussed in more detail in Chapter 4.

2. Create or obtain reagent files.

   When the synthetic approach is determined, you must then select an appropriate set of reagents, and obtain files containing the structures of the reagents. This step is discussed in more detail in Chapter 4.

3. Identify and prepare the core.

   The core is the structural element that is constant throughout the library. The side chains are built onto the core using the reagents in the reagent files. You must supply a molecule that defines the core and ensure that it is a 3D, minimized structure, using LigPrep or MacroModel, for example. This step is discussed in more detail in Chapter 4.

4. Prepare receptor and receptor grids.

   The receptor must be prepared for the docking stage, and the receptor grids for the docking must be generated. These tasks are performed in the Protein Preparation Wizard and Receptor Grid Generation panels, and are described in brief in Chapter 5 and in detail in the *Protein Preparation Guide* and the *Glide User Manual*.

5. Prepare reagent files.

   The structures in the reagent files must be converted to 3D, all-atom structures and minimized. The bonds that will be replaced when the side chains are attached to the core must also be defined. These tasks are performed in the Reagent Preparation panel, and are described in detail in Chapter 5.

6. Define attachment points and associated reagent files.

   After preparing the reagent files and the core, you define the points at which the side chains from the reagents will be attached to the core to construct the library, and associate a reagent file with each attachment point. These tasks are performed in the Define Combinations step of the Combinatorial Screening panel. This step is described in detail in Chapter 7.

7. Select parameters for the docking runs.

   Structures are docked using CombiGlide XP docking, which makes use of Glide XP technology with some variations. The appropriate Glide options for docking can be set in the Configure Docking step of the Combinatorial Screening panel. This step is described in detail in Chapter 8.

8. Define the core poses.

   A set of core poses is obtained by docking selected molecules. These core poses are used as initial positions for docking the library members. In the Define Core Poses step of the Combinatorial Screening panel you can choose which molecules to dock for the core

poses (they can be different from the one you selected above) and set constraints on the core position. This step is described in detail in Chapter 8.

9. Dock the library members.

   The docking step is done in three stages. First, the core-containing molecules are docked to generate a set of core poses. Next, structures resulting from a single substitution at any position on the core structure are docked. The unpromising reagents at each position are screened out, and the reduced list is used to dock the best of the fully substituted structures. This step is performed in the Dock Library step of the Combinatorial Screening panel, and is described in detail in Chapter 9.

10. Select the optimal library.

    With the docking stage complete, you can focus the library using the docking results to select the reagents that yield the best-scoring structures, and filter the results based on a selection of ADME properties. Finally, you can enumerate, and optionally dock, the focused combinatorial library. These tasks are performed in the Analyze Library step of the Combinatorial Screening panel. This step is described in detail in Chapter 10.

1.4 Overview of Library Enumeration

If you already have a focused set of reagents, or if you simply want to enumerate a library, you can run a combinatorial library enumeration job. You must obtain and prepare the reagents in the same way as for library design (Step 2 and Step 5); and you must prepare the core, define the attachment positions and associate the reagent files with the attachment positions (Step 3 and Step 6). The attachments are defined and the library enumerated in the Combinatorial Library Enumeration panel, which is described in Chapter 3.

1.5 Running Schrödinger Software

To run any Schrödinger program on a UNIX platform, or start a Schrödinger job on a remote host from a UNIX platform, you must first set the `SCHRODINGER` environment variable to the installation directory for your Schrödinger software. To set this variable, enter the following command at a shell prompt:

csh/tcsh: `setenv SCHRODINGER` *installation-directory*

bash/ksh: `export SCHRODINGER=`*installation-directory*

Once you have set the `SCHRODINGER` environment variable, you can start Maestro with the following command:

```
$SCHRODINGER/maestro &
```

It is usually a good idea to change to the desired working directory before starting Maestro. This directory then becomes Maestro's working directory. For more information on starting Maestro, including starting Maestro on a Windows platform, see Section 2.1 of the *Maestro User Manual*.

1.6 Citing CombiGlide in Publications

The use of this product should be acknowledged in publications as:

CombiGlide, version 2.5, Schrödinger, LLC, New York, NY, 2009.

Core Hopping

In virtual combinatorial screening, a single core molecule is used to build a library of possible ligands, by varying the side chains. The object of this strategy is to find the optimal side chains. Since in many cases it is the side chains that bind to the protein, it makes sense to vary the core also, to find other molecules ("scaffolds") to which the side chains could be attached and result in enhanced binding. This capability is available in the Core Hopping facility in CombiGlide. Core hopping requires a well-docked molecule containing an identifiable core with side chains. This molecule can come from a crystal structure, but could also come from the docking of a known active using Glide.

The core-hopping strategy is to screen multiple scaffolds (also called *protocores*) against a docked template (a lead compound), and search for alignments of potential attachment points on the scaffold with the attachment points on the template. By default, any bond to a hydrogen atom is considered a potential attachment point. However, the core molecules can instead be prelabeled to specify the attachment points. To account for the fact that a scaffold might be smaller than the template, methylene linking groups can be added to the scaffold at its possible attachment points.

The search for alignments includes both Glide docking and geometric alignment. In the first stage, the protocores (without the side chains) are aligned to the template. Glide constraints are applied to the geometrically-aligned potential attachment points. Protocores that have poor alignments are discarded. Once a set of reasonable candidate cores has been identified, the side chains are added and the fully-substituted molecules are sampled and minimized starting at their aligned positions and ranked by GlideScore. The aligned cores from the first stage can be examined and used for idea generation (see Section 2.5 on page 16). The results from the second, or docking, stage can be examined for possible use as synthetic leads or as starting points to CombiGlide's Combinatorial Screening workflow, which can be used to optimize the side chains on the newly discovered core.

We have already introduced the concept of a *template*, a docked molecule with a defined core region. We use the term *protocore* to denote a candidate core molecule that is aligned with the template in the first stage of core hopping, defined above. Different alignments of a protocore can then result in multiple *cores*, which consist of the protocore molecule with a specified set of attachments corresponding to those on the template. Finally, we use the term *raw core* to denote a molecule provided to the Protocore Preparation procedure as input. The process can then be described as follows: raw cores are supplied to the Protocore Preparation procedure, which produces protocores; protocores are aligned with the template to produce cores.

2.1 Defining Attachment Bonds

Potential attachment points are labeled in a structure by defining “grow bonds”, or “attachment bonds”. These are specially labeled bonds that Maestro uses to grow structures, by automatic replacement of the structure at one end of the bond with another structure. Regular structure files do not have grow bonds defined, so you must add them. We use the term *raw core* for these molecules to make it clear that you will be adding editing directives for the grow bonds to these molecules before they are turned into protocores by the Protocore Preparation procedure.

The default operation of the Protocore Preparation stage is to consider all bonds to hydrogens or all nonpolar bonds to hydrogens as potential attachment bonds. If you are satisfied with the defaults, or if you have structures that are already labeled with grow bonds, you can proceed to protocore preparation. If you want to use grow bonds other than the defaults, you can define them using the Edit Raw-Core Attachment Bonds panel, which you open from the Core Hopping submenu of the CombiGlide submenu on the Applications menu.

The Edit Raw-Core Attachment Bonds panel allows you to add the defaults to a molecule if you wish, and to clear all attachment bonds. Its main function is to pick individual attachment bonds in the Workspace.

For terminal atoms, picking adds an bond if it is not already marked, and clears an attachment bond if it is marked. For non-terminal atoms, picking an unmarked bond adds it with the default direction, clicking again changes the direction, and clicking a third time clears the attachment bond. Clicking on a marked bond changes it to the next state in this cycle.

As each new raw core is brought into the Workspace, the editing changes made to the previous raw core are saved.

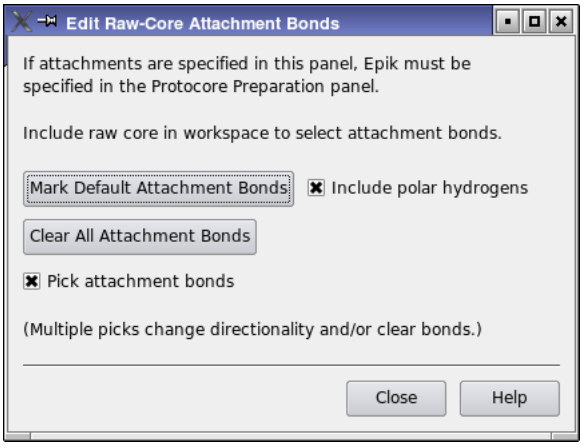


Figure 2.1. The Edit Raw-Core Attachment Bonds panel.

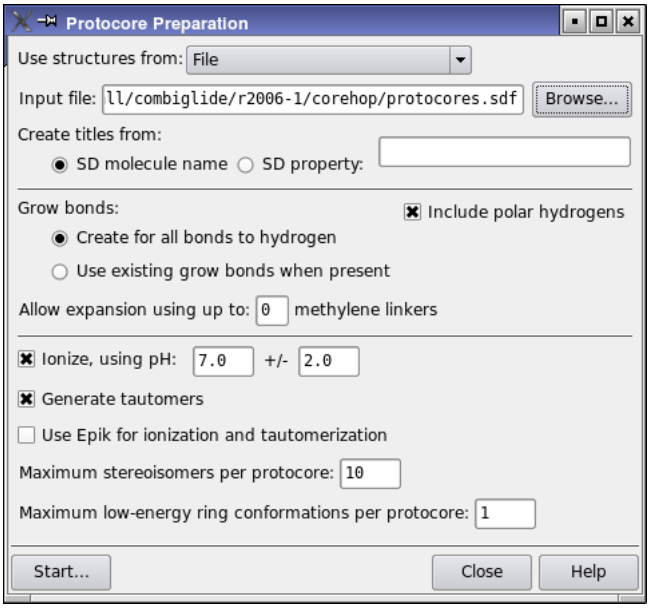


Figure 2.2. The Protocore Preparation panel.

2.2 Protocore Preparation

In addition to any structural preparation needed to generate an all-atom, 3D structure, the protocore molecules must be labeled with the potential attachment points. These points are specified in terms of grow bonds, just as is done for reagent preparation. This task is done in the Protocore Preparation panel, which you open from the Core Hopping submenu of the CombiGlide submenu on the Applications menu. As for reagent preparation, protocore preparation uses LigPrep for the preparation tasks. For more information on LigPrep, see the *LigPrep User Manual*.

A library of protocores in both unprepared and prepared form is available from our web site, http://www.schrodinger.com.

The Protocore Preparation panel is divided into three sections, which are described in the following subsections. In the top section, you specify the source of the protocore structures, and in the bottom section you select options for structure preparation using LigPrep. These two sections are the same as in the Reagent Preparation panel.

2.2.1 Choosing the Source of Protocore Structures

In the top section of the panel, you can choose the source of structures from the Use structures from option menu. You can select structures from the Workspace, the Project Table, or from an external file. Structures from the Workspace or the Project Table are written to a Maestro file that is passed to the protocore preparation job.

If you choose File from the Use structures from option menu, you can enter the path to the file in the Input file text box, or click Browse to navigate to the file. The file must be in Maestro format or SD format. The structures can be 2D or 3D, and are converted to all-atom 3D structures by the protocore preparation job.

If you selected an SD file, you can also specify the source of the protocore titles. Titles are used for identification in CombiGlide, so it is important to have unique titles. If you do not make them unique, CombiGlide will do so for you. You can select SD molecule name to use the molecule name from the SD file, or select SD property and enter a property name in the text box to use the property for the titles.

2.2.2 Defining Attachment Points

There are three ways to specify which bonds of the raw cores supplied as input for Protocore Preparation are to be considered attachment points. First, you can use the defaults. Second, you can use a selection of attachment points previously made in the raw-core editor, as described in Section 2.1 on page 8. Finally, if you are downloading data from a database in the form of an SD file, you can use a special annotation in the SD file to denote the attachment positions. Any terminal atom given the atom type `R#` in an input SD file is used by Protocore Preparation to define an attachment. If an input molecule has any such atom types, only these atoms will be labeled as attachment points in the resulting protocore.

You can make the choices for the attachment points in the middle section of the Protocore Preparation panel.

If you only want to consider nonpolar bonds when creating grow bonds, deselect Include polar hydrogens.

If you do not have labeled structures, select Create for all bonds to hydrogen.

If you have structures that are already labeled with grow bonds or you are reading an SD file with `R#` atom types, or have a mixture of structures with and without labeling, select Use existing grow bonds when present. Grow bonds will then be created for structures that do not have them, subject to the Include polar hydrogens option.

If you want to add methylene linkers to your protocore molecules to extend their size, enter the number of linkers in the Allow expansion using up to *N* methylene linkers text box. A small protocore might only match the template with linkers whereas a large protocore might match without linkers. If linkers are present in the input protocore file, they are used only if needed during protocore docking. Inserting linkers generally increases the number of protocores returned. We do not recommend the specification of more than two linkers. Using linkers expands the number of ways protocores can match the template, but also slows down the Stage 1 matching process by a factor of about four. Without linkers, Stage 1 takes approximately 5 seconds per protocore on a 3 GHz processor.

2.2.3 Setting Options for Structure Variation and Cleanup

In addition to defining the attachment points of the protocores, the preparation process performs a 2D-to-3D conversion, generates all reasonable ionization and tautomeric states, and expands the stereochemistry. If the structures you have are already 3D structures in the appropriate ionization state, tautomeric and stereochemical form, you might not need to generate variations. Many collections of structures are provided in 2D form, with implicit hydrogen atoms. To generate 3D, all-atom structures for docking, some variation to obtain structures that represent the actual state under physiological conditions may be needed.

The lower portion of the Protocore Preparation panel allows you some control over the structure variation and cleanup process, by setting options for the following:

- Ionization state—You can request that the structure be protonated or deprotonated to represent the actual forms in water in the given pH range, by selecting Ionize, using pH and entering the pH value and range in the accompanying text boxes. If this option is not selected, the ionization state is left as is.
- Tautomerization state—You can request the generation of tautomers of the input structures by selecting Generate tautomers. Ionized structures might have a different tautomeric form from the un-ionized structures.
- Use of Epik for ionization and tautomerization—Epik is a more sophisticated tool for generating ionization and tautomerization states than the default tools provided with LigPrep, the `ionizer` and the `tautomerizer`. It is separately licensed, so you must have an Epik license to use this option. You *must* use Epik if you have defined explicit attachment points.
- Stereochemistry—Stereoisomers of the input structures are generated in the absence of chirality information, either explicitly specified or deducible from the 3D structure. If the input structures are 2D and have chiral centers but do not have chirality information, you should ensure that the number specified in the Maximum stereoisomers per protocore text box covers the possible combinations.

- Ring conformations—If the input structures contain rings that can exist in more than one low-energy conformation, enter the number of likely low-energy conformations in the Maximum low-energy ring conformations per protocore text box. Ring conformations are not searched during docking, so the conformations must be set up during the reagent preparation process.

If you do not need to generate structural variations, you should deselect Ionize, using pH and Generate tautomers, and enter 1 in the Maximum stereoisomers per protocore and Maximum low-energy ring conformations per protocore text boxes.

2.3 Template Preparation

Before you proceed to the docking stages, you must prepare a template molecule, for example from a lead compound. Template preparation can be done in the first step of the Combinatorial Screening panel. Follow the instructions in Chapter 7, and save the library definition as described in Section 7.3. The library definition is used as input to the protocore docking to define the template.

It is not necessary to specify reagent files at the attachment points on the template molecule. However, if reagent files are specified, it is much easier to carry promising cores that are identified in the core hopping workflow through the combinatorial screening procedure, which will optimize side chains for the new core. To avoid adding reagent files, it is necessary to click Cancel in the reagent-file browser, and thus not specify a reagent file. This action does not prevent you from saving the definition file.

Since one of the purposes of core hopping is to rigidify a flexible scaffold present in a lead compound, we recommend defining the core of the lead compound to be rather large (including linker atoms, should there be any). Then the core-hopping procedure will attempt to find structures that span the set of attachment points specified.

2.4 Protocore Docking

Once you have prepared your protocores and template, you can proceed to the alignment and docking of the protocores. This can be done in the Protocore Docking panel, which you open from the Core Hopping submenu of the CombiGlide submenu on the Applications menu. The panel is divided into four sections: protocore structure selection, stage selection, template selection, and receptor selection, which are described in the following subsection. When you have finished making selections, click Start to open the Start dialog box in which you can make job settings and run the job, or click Write to write the input files. See Chapter 2 of the *CombiGlide Quick Start Guide* for a tutorial example of protocore docking.

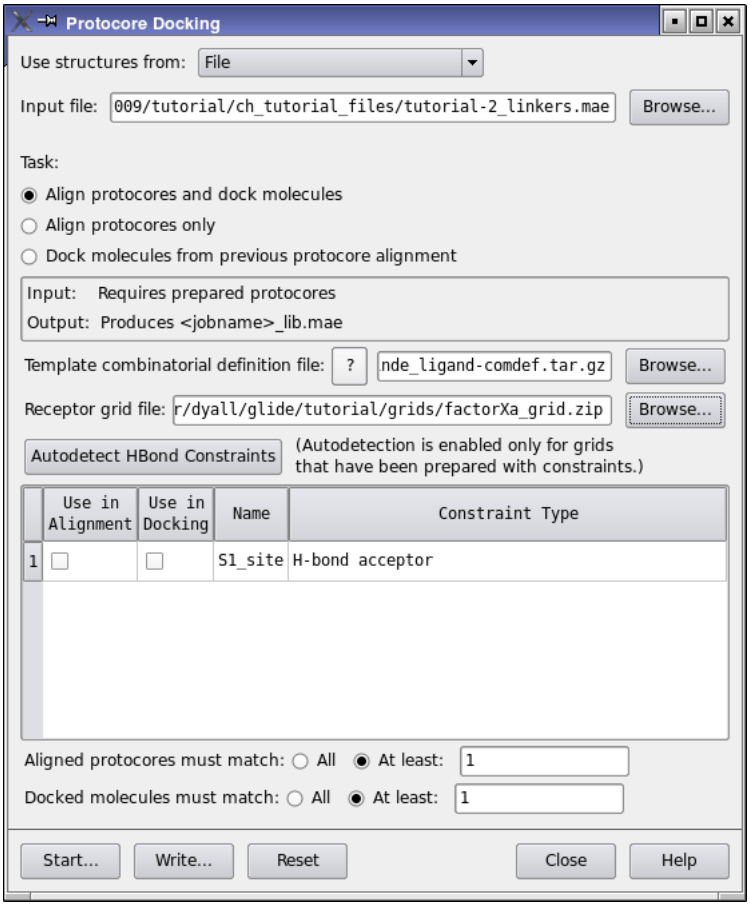


Figure 2.3. The Protocore Docking panel.

2.4.1 Protocore Structure Selection

At the top of the panel, you can select the source of protocore structures from the Use structures from option menu. If you have just run Protocore Preparation and incorporated the results, the protocores should be in the Project table as an entry group, and you can choose Project Table (selected entries) from the Use structures from option menu.

Otherwise, you can choose the Workspace structure, or read structures from a file. To read structures, choose File from the Use structures from option menu, and enter the file name in the Input file text box, or click Browse and navigate to the file. The file should be a Maestro file. If you are looking for the structures from a protocore preparation job, the file should be named *jobname*`-out.mae`.

If you are only performing the second docking stage, the input file you choose should not be the protocore preparation output file, but a file in which the protocores have been "dressed" with the side chains. This will normally be the *jobname*_dresssed_lib.mae file that results from a previous Stage 1 protocore docking job.

2.4.2 Task Selection

In the Task section, you can choose which of the docking stages to perform.

In Stage 1, the protocores are docked with constraints to the template attachment positions. During docking, each protocore can give rise to zero, one, or more cores, where a core corresponds to a protocore plus a selection of its attachment bonds that match those of the template. If the protocore does not have enough attachment points to align to all those on the template, the program will align the number on the protocore.

The cores are written to a file without any attachments; a second file is generated in which each core is "dressed" with the side chains of the template, grown from the attachment positions, which can be used in Stage 2.

In Stage 2, the dressed cores are docked to the receptor and ranked by GlideScore.

- If you want to examine the results of protocore docking and select protocores before proceeding to the docking of the dressed cores, select Align protocores only. When you have the results, use the dressed structures and select Dock molecules from protocore alignment.
- If you want to dock all the possible cores that arise from your set of protocores, select Align protocores and dock molecules. Both docking stages are run as part of the same job.

2.4.3 Template Selection

The template must be already prepared, according to the procedure given in Section 2.3 on page 12. The template is defined by a library definition file, which is a gzipped tar file with the extension -comdef.tar.gz. You can of course override the extension when you save the library, but it is preferable to leave it so that you can distinguish this file from other gzipped tar files. To select the template, enter the name (including the path if necessary) into the Template combinatorial definition file text box, or click Browse to navigate to the file. For information on how to prepare the template, click the query button to the left of the text box. An example is given in Section 2.4 of the *CombiGlide Quick Start Guide*.

2.4.4 Receptor Selection

In the Receptor grid file section of the panel you can select the receptor grid that you want to use for docking the protocores and dressed cores. The receptor grid must be prepared beforehand, using the instructions in Chapter 4 of the *Glide User Manual*. You can specify either a `.grd` file, if the grid files are not in a zip archive, or you can specify a `.zip` file that contains a grid. To select the grid, enter the name (including the path if necessary) into the Receptor grid file text box, or click Browse to navigate to the file.

2.4.5 Constraints

Protocore alignment and docking can be constrained to meet conditions for hydrogen bonds. If the receptor grid contains constraints, these are added to the constraints table in the lower part of the panel. The table has four columns, the last two of which list the constraint name and the constraint type. The first two columns allow you to select a constraint for use in the alignment stage or in the docking stage, by providing a button that you can click on. Only H-bond constraints can be selected: there are no buttons provided for other kinds of constraints.

You can automatically select H-bond constraints that are satisfied by the template, by displaying the template and the receptor in the Workspace, and clicking Autodetect H-bond Constraints. If a constraint is met by a template core atom, it is turned on for both alignment and docking. If a constraint is met by a template side-chain atom, it is turned on for docking only. Alignment input does not contain side chains, so constraints on side chains will not, in general, be met. When you click this button, a dialog box opens in which you can set the parameters that define a hydrogen bond. On clicking OK, the selections are made in the constraints table. You can subsequently make changes to the selections if you want.

The conditions that apply to Glide constraints also apply here—see Section 5.5 of the *Glide User Manual*. For example, a maximum of 4 constraints is permitted for each stage. Even within the permitted limits, specifying too many constraints could result in elimination of all cores in the library. You can select a minimum number of constraints to satisfy, both for protocore alignment and docking, by selecting the appropriate At least option and entering a value in the text box. The default is to satisfy all constraints.

You can also turn on H-bond constraints that are not satisfied by the template. This possibility allows you to select cores that interact with the receptor in a way that the template does not. However, unless the protocore library is large enough to contain multiple heterocyclic variations of cores, it might be better to run with few or no core constraints and to examine promising aligned cores for locations where heteroatoms might be added to make desired interactions.

2.4.6 Input and Output Files

Input files are written when you click Start or Write. In addition to the input files, clicking Write writes a shell script named *jobname*`.sh`, which you can use to run the job. The shell script has a `-help` option that describes how to set job options.

The structural output of Stage 1 is a set of cores whose selected attachment bonds are aligned to those of the template. Two files containing core structures are written: a file that contains the "bare" core structures, without any side chains, and a file that contains the "dressed" core structures, with the side chains of the template attached to the attachment points that match the template. This second file is named *jobname*`_dressed_lib.mae`.

The structures in this file are sorted by two keys (properties). The primary key is `cgch nmatch`, which is the number of attachment bonds in a core that were aligned to the template. The secondary key is `cgch geometry`, which is a measure of the "goodness of fit" of the core to the template.

The structural output of Stage 2 is the set of docked dressed cores ranked according to GlideScore. It is written to a file named *jobname*`_lib.mae`.

2.5 Examining the Results of Core Hopping

You can visualize the results of core hopping in the Workspace with the help of the Core Hopping Visualization panel, which you open from the Core Hopping submenu of the CombiGlide submenu on the Applications menu. This panel facilitates visualization of the Stage 1 (aligned) or Stage 2 (docked) output structures, aligned with the template, along with the surface of receptor. It also places the properties of interest to the left in the Project Table and hides the other properties. The Project Table view can be reverted to the default either by closing the panel or by clicking a button in the panel.

To use the panel:

1. Import the template, receptor, and aligned or docked cores into Maestro.

   The aligned cores are in the file *jobname*`_dressed_lib.mae`; the docked cores are in *jobname*`_lib.mae`. You can visualize either of these sets of structures.

2. Clear the Workspace.

   You can do so by clicking this toolbar button:

3. Open the Core Hopping Visualization panel.

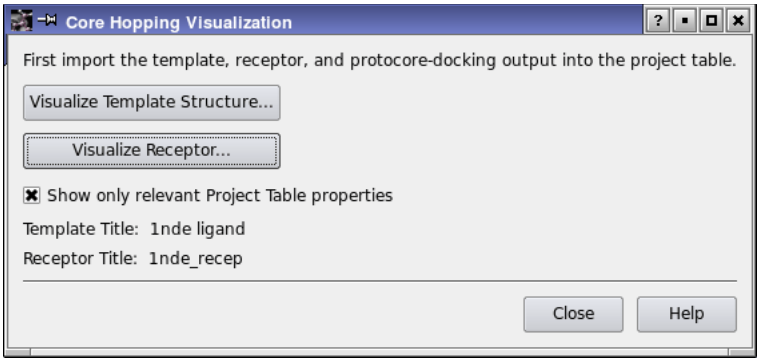


Figure 2.4. The Core Hopping Visualization panel.

4. Select the template in the Project Table, and click Format Template Structure For Core Hopping Viewing.

   A dialog box is displayed that provides instructions and a button to click when the template is selected. When you click the button, the template is displayed. The grow bonds are marked on the template, the carbons are colored pink, and the template molecule is "locked" in the Workspace.

5. Select the receptor in the Project Table, and click Format Receptor For Core Hopping Viewing.

   A dialog box is displayed that provides instructions and a button to click when the receptor is selected. When you click the button, a receptor surface is created, and the Surface Table panel is opened. The receptor entry is "locked" in the Workspace, but only the surface is displayed. You can close the Surface Table panel.

6. Include any of the poses in the Workspace to view them superimposed on the template.

The structures in the *jobname*`_dressed_lib.mae` file are "dressed" with the side chains from the template molecule, using simple geometric criteria. These structures may not be optimally docked: for example, the side chains could clash with the receptor. The side-chain alignment is not the main intended purpose of viewing these structures: rather, it is to identify promising cores and generate ideas such as locations for heteroatom substitutions that are likely to improve the interaction with the receptor.

Here are some suggestions for viewing the results:

- Sort the Stage 2 results by the Stage 1 sorting order, to see the correlations.

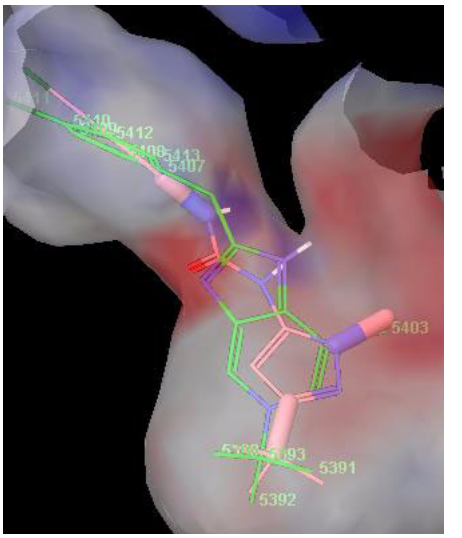


Figure 2.5. Workspace showing template and core pose with the receptor surface.

- Sort the results by the vector_min property. This property indicates how well the direction of the most poorly aligned attachment bond of a core candidate matches that of the corresponding attachment bond on the template. This is a dot product, so higher numbers indicate better alignment.

The cores that align well to the template can be examined with an eye toward synthesis or toward chemical modification to improve binding.

Since we supply only a small protocore library, it is possible that Stage 1 will find scaffolds that align well with the template but whose interaction with the receptor can be improved by means of heteroatom substitution.

The properties that relate to core hopping are described in Table 2.1. These properties are displayed in the Project Table when you open the Core Hopping Visualization panel, and other properties are hidden.

Table 2.1. Core-hopping properties.

Property	Description
cgch nmatch	The number of attachment bonds in a core that were aligned to the template. This is generally equal to its maximum possible value—namely, the number of attachment bonds in the template—but it would be less if the originating protocore had fewer possible attachment bonds than this.
cgch geometry	Measure of alignment, computed in a similar way to an atomic RMS displacement between the atoms at the ends of the template's attachment bonds and the corresponding atoms on the aligned cores. The atoms used in the calculation are displaced by a standard 2 Å, which serves to remove variations in bond length and weight the results towards common bond-vector directionality.
cgch nlinker	The number of linkers actually used to align this core. This property is equal to the sum of the number of linkers in all attachment bonds on the aligned core. Lower is better, since lower gives fewer rotatable bonds; but sometimes adding linkers is the only way to achieve reasonable "cgch geom" values
cgch from rms	This is equal to the interatomic RMS between the "From" atoms on the template and the corresponding atoms on the aligned core. The values are taken from the alignment stage, even for Docking output. Lower is better.
cgch vector avg	Scalar product of the vectors defined by the attachment bond on the template and that on the aligned core, averaged over all attachment bonds. Higher is better.
cgch vector min	Minimum of the scalar products of the vectors defined by the attachment bond on the template and that on the aligned core. This property measures how bad the worst bond vector alignment is. Higher is better.
cgch core	The unadorned protocore name. The title includes appended information about which bonds were aligned.
cgch coreindex	The sequential index of the protocore in the original protocore file.
outrank rms vector	Difference between the number of structures this structure outranks and the number of structures that outrank it. Outranking means that structure has better values of both "cgch from rms" and "cgch vector min". Not displayed by default in the Project Table. Higher is better.
pareto rms vector	Boolean property that is true if there is no other structure that has both a better value of "cgch from rms" and a better value of "cgch vector min". Not displayed by default in the Project Table.

Enumerating a Combinatorial Library

CombiGlide provides tools for simple library enumeration in addition to the tools for virtual combinatorial screening and core hopping; that is, for creating a library of ligands from a core structure to which side chains are added in user-specified positions. The library can also be docked with Glide. There are two tools for this task, the Interactive Enumeration and Docking panel, which allows you to interactively select and prepare both the core-containing molecule and the reagent molecules (fragment collection) for the library, and the Combinatorial Library Enumeration panel, which allows you to prepare the core-containing molecule but uses predefined reagent libraries to create the library. Both of these panels can be opened from the CombiGlide submenu of the Applications menu in the main window.

3.1 Interactive Enumeration and Docking

The Interactive Enumeration and Docking panel provides tools for all the tasks necessary to set up, enumerate, and dock a combinatorial library. The panel provides the means to define the attachment points on the core-containing structure and to set up the libraries for each attachment point. For each attachment point, you can also include methylene linkers between the core and the fragment that is attached. You can select structures from a predefined fragment library or create a new fragment library from a set of 3D structures. Unlike the combinatorial screening workflow, the entire combinatorial library is enumerated and, if desired, docked. Thus, this facility is not intended for use with very large combinatorial libraries.

To open the Interactive Enumeration and Docking panel, choose Applications > CombiGlide > Interactive Enumeration and Docking in the main window. From Linux, you can open this panel as an independent application with the following command:

```
$SCHRODINGER/utilities/cg_interactive_enum_dock
```

The panel has two tabs in the upper section, the Combinatorial Library tab and the Fragment Collection tab.

3.1.1 Setting Up the Combinatorial Library

To set up a combinatorial library, you need to choose a core-containing molecule, define its attachment points, and add a fragment collection at each attachment point. These tasks are done in the Combinatorial Library tab. You can use an existing library and modify it, or you can create a new library.

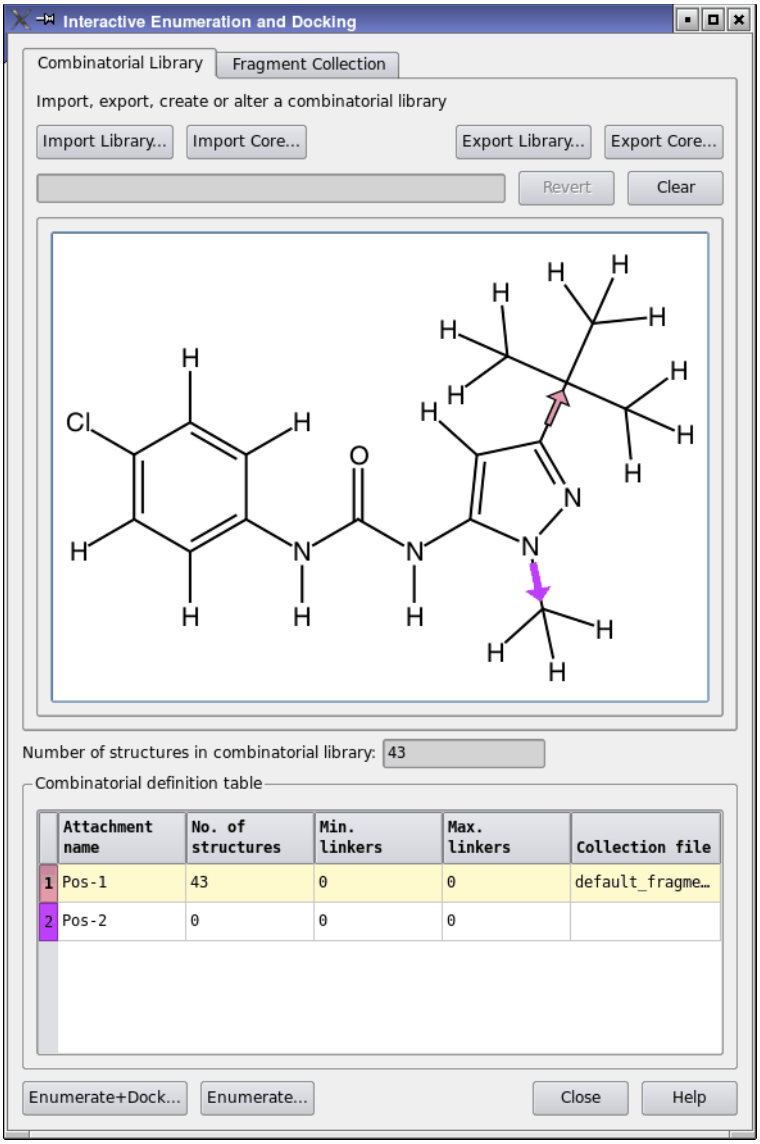


Figure 3.1. The Interactive Enumeration and Docking panel, Combinatorial Library tab.

If you already have a library, you can import it by clicking Import Library. You are prompted to save or discard changes to the current library if they are not saved, then a file selector opens, in which you can navigate to an existing library, which has the suffix `-comdef.tar.gz`. The library is loaded, the core molecule is displayed with its attachment points, and the Combinatorial definition table is filled in.

If you want to create a new library, the first step is to import a core-containing molecule. To do so, click Import Core. You are offered a choice[1] of importing from the Workspace, the Project Table, or a file. If you choose Import from Workspace, there must be a single entry in the Workspace. Likwise if you choose Import from Project Table, there must be a single entry selected in the Project Table. If you choose Import from File, a file selector opens, in which you can navigate to and select the Maestro file that contains the structure you want to use to define the core. The first structure in the file is read.

When you have imported the core-containing molecule, it is displayed in the display area with any existing attachment bonds, or core positions, marked by colored arrows. The arrow color matches the rows in the Combinatorial definition table. The head of the arrow indicates the fragment or atom that will be removed when a fragment (R group) is attached in this position.

You can then pick bonds in the structure to define the attachment points, or core positions. If you right-click on an existing core position, the shortcut menu allows you to switch the direction of the arrow or delete it. Clicking again on the bond also changes the direction of the arrow. The direction is only changed, however, if it would not invalidate existing core positions. Each time you create a new core position, it is added to the Combinatorial definition table. The selected position is highlighted in the table and the arrow is outlined in black in the display area.

To delete a core position, right-click on the arrow in the display area or in the table row and choose Delete.

The Combinatorial definition table lists each core position, along with information about the position and its attachments: the number of structures in the fragment collection used at this position, the minimum and maximum number of methylene linkers, and the name of the fragment collection file. The fragment collection itself for a given core position is shown in the Fragment Collection tab, which is described in the next section.

When a core position is first selected, a default name is created and the numbers in the columns are set to zero. You can edit the name by clicking in the table cell and changing the text. The fragment collection for this position in the Fragment Collection tab is initially empty. To associate a fragment collection with this position, you can create one in the Fragment Collection tab, or you can read one from file.

To read a new collection from file, right-click on the selected row of the table and choose either Replace Collection or Append to Collection. It does not matter which you choose because the existing collection is empty. In the file selector that opens, you can navigate to and select a fragment collection, which has an extension of `.bld`. The file selector opens by default in the Schrödinger Collections directory, which supplies a default set of fragments. The directory

1. If you opened the panel from the command line, there is no menu choice, but a file selector opens immediately.

selection also includes My Collections, which is set to the `cg/interactive_fragments` subdirectory of your user resources directory (`$HOME/.schrodinger` on Linux, `%APPDATA%\Schrodinger` on Windows) as a convenient location for storing your fragment collections. Once the collection is read in, it is shown in the Fragment Collection tab. The number of fragments in the file is displayed in the No. of structures column.

If you have created a fragment collection in the Fragment Collection tab, you can apply it to the current core position by right-clicking and choosing Apply Fragment Collection, which replaces the current collection with the entire collection, or choosing Apply Selected Fragments, which replaces the current collection with the fragments that are selected in the Fragment Collection tab. A default name for the fragment file is created. You can change it by right-clicking on the table row and choosing Rename.

As well as the fragments that you define, the combinatorial library can include methylene linkers between the core and the fragment. The number of linkers is determined by the values given in the Min. linkers and Max. linkers columns, which you can edit to set the desired values.

If you have changed a library definition and want to revert to the original definition, click Revert. All changes since the last time the library was saved are discarded and the library is reloaded.

To start a new library definition, you can click Clear to clear the current library definition.

When you have a library that you want to save, click Export Library. The library must have a fragment collection associated with each core position. A file selector opens, in which you can navigate to the desired location and name and save the library definition.

3.1.2 Setting Up a Fragment Collection

The Fragment Collection tab provides the tools to design your own fragment collection. You can import fragments, create fragments, delete fragments, and export fragments. You can either set up a fragment collection independent of a core position, or you can set up a fragment collection for a particular core position. If a core position already has a collection file associated with it, the contents of this file are loaded as the current collection when you select the core position.

To add fragments to a collection, you can import them or create them. Creating new fragments is done by clicking Create and working in the Create Fragment Collection panel. This task is described in the next section. In addition to using the Create Fragment Collection panel, you can prepare fragments in the Reagent Preparation panel and then import them—see Section 5.2 on page 38 for more information.

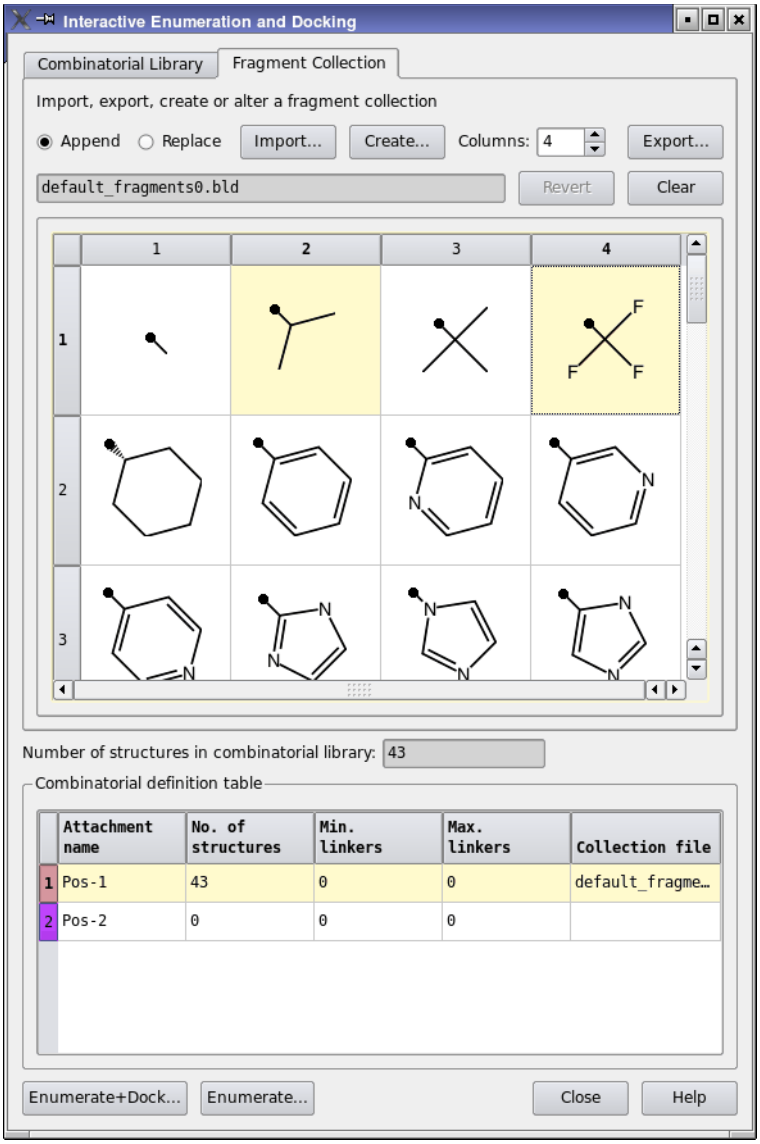


Figure 3.2. The Interactive Enumeration and Docking panel, Fragment Collection tab.

You can import fragments from a file, from the Project Table, or from the Workspace. The fragments must be already prepared with the information on the attachment bond, otherwise you will not be able to import them. If you import from a file, the file must have a `.bld` extension.

Before you click Import to import the structures, you should decide whether you want to add the structures to the existing collection, by selecting Append, or replace the existing collection with the new structures, by selecting Replace.

The 2D structures of the current fragment collection are displayed in the center of the tab. The filled circle indicates the connection point to the core. You can set the number of columns of structures in the display area in the Columns box. You can select multiple structures by dragging or with shift-click and control-click. When you right-click on a structure, the shortcut menu offers the choice of deleting this structure or the selected structures, or retaining only this structure or only the selected structures.

If you are editing the fragment collection for a particular core position, the collection that is used for that position is not changed unless you specifically change it. This means that you can experiment with the collection before you commit to applying it to the core position. When you want to apply the changes, you can choose one of the following options from the shortcut menu of the Combinatorial Definition table

- Apply Fragment Collection—replace the collection file with the entire fragment collection
- Apply Selected Fragments—replace the collection file with the fragments that are selected in the fragment display area.

If you want to discard the changes you have made to an existing collection, click Revert. The changes are discarded and the structures are read in again from the file. If you want to clear all the fragments from the current collection, click Clear. In both cases you are prompted to confirm the action.

When you want to save a collection, you can click Export to export it to a file or to the Project Table as an entry group. A menu is displayed so that you can choose the destination. The default file location is My Collections, which is described in the previous section.

3.1.3 Creating a Fragment Collection

If you want to create a new fragment collection by selecting from a set of structures, you can do so in the Create Fragment Collection panel. In this panel, you can import structures, select the ones that you want, define their attachment bonds, and add them to the current fragment collection in the Interactive Enumeration and Docking panel.

To open the Create Fragment Collection panel, click Create in the Interactive Enumeration and Docking panel.

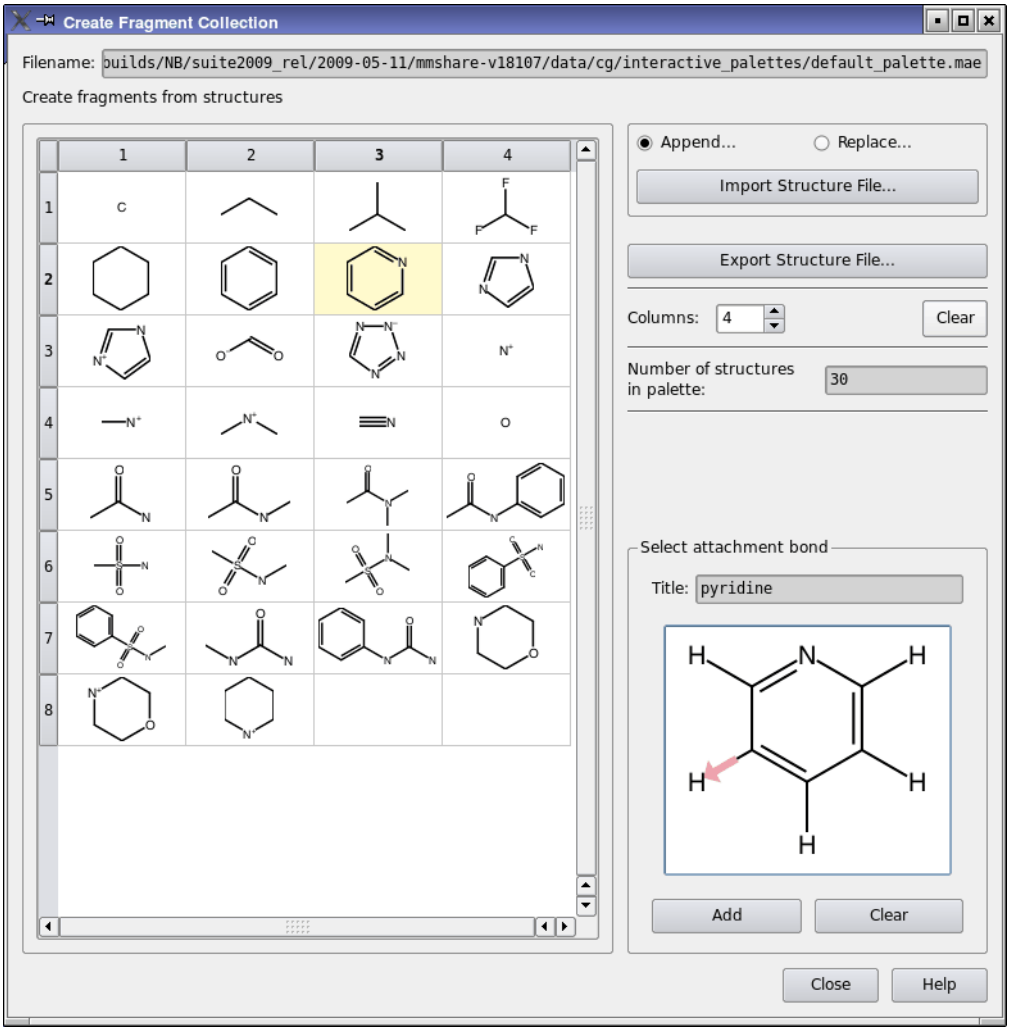


Figure 3.3. The Create Fragment Collection panel.

The palette—the display area for the structures—is initially empty. The palette is a table of structures that can be selected for inclusion in the fragment collection, with an attachment bond defined. You can set the number of columns in the Columns box, and resize the table cells by dragging the row or column boundary in the table margin. You can clear the palette by clicking Clear.

To add structures to the palette, click Import Structure File. A menu is displayed that allows you to choose whether to import from a file, the selected entries in the Project Table, or the

Workspace. If you choose Import from File, a file selector opens, and you can navigate to and select the file. The file can be a Maestro file, without attachment bonds defined, or a reagent file with attachment bonds defined (`.bld`). The file selector opens in a directory that contains a default set of structures that are useful in a palette. When you click Open, the file name is displayed in the Filename text box, and the structures are shown in 2D form in the palette.

If you already have structures, you can append to the list by selecting Append and importing structures. If you want to replace the existing structures with a new set, select Replace before importing the structures.

You can select multiple structures in the palette (with shift-click and control-click). The shortcut menu offers the choice of keeping only the selected structures or deleting them, or keeping only the structure clicked on or deleting it. This allows you to define new palette collections for later use.

If you want to save a particular set of structures, click Export Structure FIle. The menu that is displayed offers the choice of exporting to a Maestro file or to the Project Table. Attachment bond information is not written with the structures. If you choose to export to a file, a file selector opens, in which you can navigate to a location and name the file. The file selector opens in a location beneath your home directory that can be used as the default location of any structure files that you might like to save for reuse in a palette. The export option is useful for creating a new file for use future use in a palette after importing and possibly deleting structures from several files.

Once you have structures in the palette, you can proceed to select a structure, add an attachment bond, and then add the structure to the current fragment collection in the Interactive Enumeration and Docking panel.

When you click on a structure in the palette, the structure is displayed in the Select attachment bond section, so that you can define the attachment bond. The title of the structure is displayed in the Title text box, and the structure itself in the display area. To select the attachment bond, click on a bond in the display area . Clicking a second time on the same bond changes the direction of the attachment bond, if the bond is not terminal. The arrow on the bond points towards the part that is removed to attach the fragment to the core.

To add this fragment to the collection, click Add. The fragment flashes, and it is added to the current Fragment Collection display of the Interactive Enumeration and Docking panel. It is therefore a good idea to have both panels visible and next to each other, so that you can see the fragment being added to the fragment collection.

3.1.4 Enumerating and Docking the Library

When you have defined a combinatorial library, you can then proceed to enumerate it, and if you want, to dock it.

If you click Enumerate, the Start dialog box opens, and you can set job options and start the enumeration job.

If you click Enumerate+Dock, the Grid File and Glide Precision dialog box opens first. In this dialog box, you can select the Glide precision, from SP, XP, or HTVS, and specify the grid by clicking Browse and navigating to the grid file. When you click OK, the Start dialog box opens, and you can set job options and start the enumeration and docking job.

The results can be appended to the Project Table or written to the current directory.

3.2 Combinatorial Library Enumeration

The combinatorial library enumeration is carried out by substitution of side chains on the core structure with fragments from the reagent structures. For both the core and the reagents, the bonds to be broken must be designated. The fragments from the reagents are then attached to the core at the site of the broken bond. This process can be regarded as "growing" the new side chain onto the core in place of the old side chain, and the bond that is replaced (with its attached atoms) is called the grow bond. These bonds are not necessarily the bonds that are broken and formed in the real chemical reaction, but represent a means of varying the side chains on a core structure that might include part of the real reagent.

To generate a combinatorial library, you must have a 3D, minimized structure for the core-containing molecule whose side chains you want to replace, and a set of prepared reagent files that define the replacements. For information on preparing the core structure and the reagent files, see Chapter 5.

The Combinatorial Library Enumeration panel (Figure 3.4) is essentially the same as the Define Combinations step of the Combinatorial Screening panel, which is described in Chapter 7. The procedure is summarized here.

When you first open the Combinatorial Library Enumeration panel, Pick molecule is selected in the Core structure section, and most other controls are unavailable. After you have selected a core-containing molecule, the other controls become available.

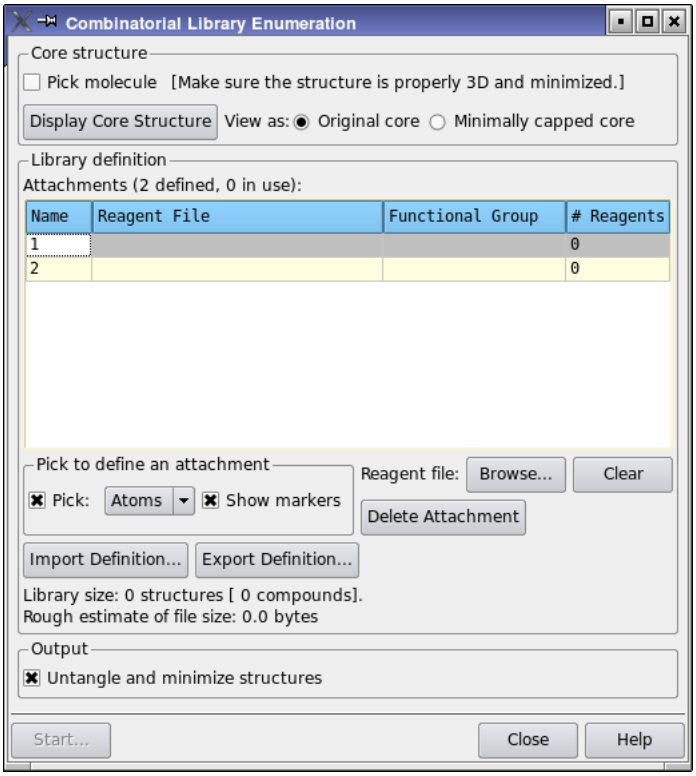


Figure 3.4. The Combinatorial Library Enumeration panel.

To define the core-containing structure:

1. Ensure that the structure is displayed in the Workspace.
2. Click on an atom in the core-containing molecule.

   The core-containing molecule is now defined, and you can define the attachments. Pick molecule is deselected, and the Pick option is selected in the Pick to define an attachment section.

To define an attachment:

1. Pick two atoms that define the position of the attachment, the atom that is kept first, then the atom that is deleted.

   A turquoise arrow is displayed over the bond, and the Select Reagent File dialog box opens.

2. Select the appropriate reagent file, and click OK.

   The reagent files have a `.bld` extension: they are Maestro-format files that have special information for the Maestro builder, which does the library enumeration. The dialog box closes, and details of the attachment are listed in the Attachments table. The arrowhead changes to a sphere and the color changes to gold. The size of the library with the current set of attachments is given in the lower portion of the panel.

If you choose the wrong location for an attachment position, you can delete it by selecting it in the table and clicking Delete Attachment. If you attach the wrong file, you can select the attachment position in the table, and click Browse (below the table) to select the correct file.

When you have defined an attachment, the size of the resultant library is displayed at the foot of the Library definition section.

It is recommended that you leave the Untangle and minimize structures option selected. Untangling ensures that (for example) any chains that pass through the middle of rings during the build process are removed from the rings and the structures are properly minimized. However, if you are not interested in having accurate structures, you can deselect this option.

To start the library enumeration job:

1. Click Start.

   The Start dialog box is displayed.

2. Select job options, and click Start.

The host you choose should have temporary storage space of about 3 to 4 times the size of the library. This space is required for the untangling and minimization of the library members.

The job takes about 45-60 seconds per output structure when each structure is minimized and untangled. The library is written to a Maestro file named *jobname*`.mae`.

Combinatorial Screening: Defining the Chemistry

The first step in the design of a focused library using CombiGlide is to identify the chemistry associated with desired library. Most likely, this will be done in collaboration with one of your medicinal or combinatorial chemists.

For example, if you were interested in generating a focused pyrazole library in an attempt to identify inhibitors of p38 MAP kinase, your synthetic chemists might suggest the approach described in Figure 4.1.

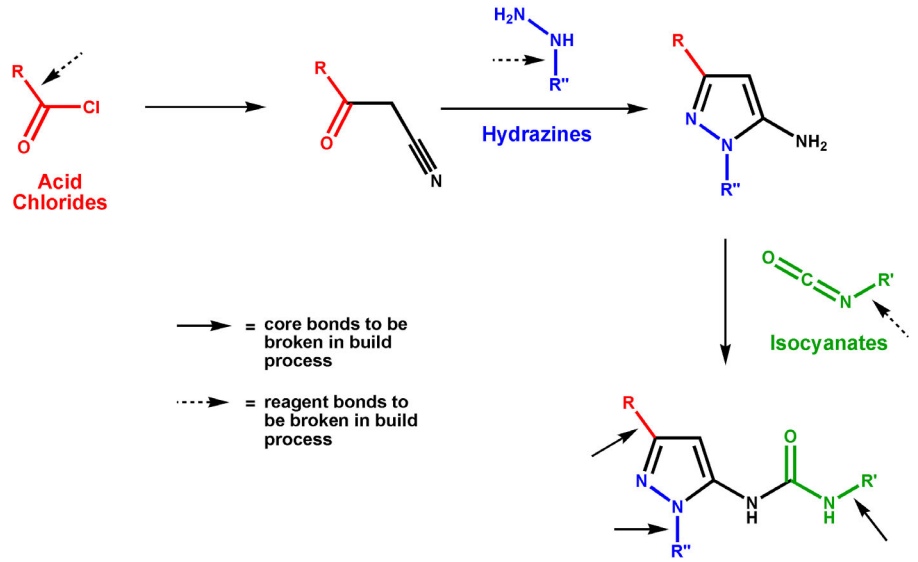


Figure 4.1. Synthetic route to pyrazole library.

4.1 Obtaining Reagent and Protein Structure Files

Once the synthetic approach has been determined, it will be apparent which types of reagents will be required for the library synthesis. For example, in the pyrazole library described in Figure 4.1, the library will be generated by the combination of acid chlorides, hydrazines, and isocyanates.

You must then obtain files that contain the structures of the reagents that you want to use. For each reagent, the structures must be contained in a single, multistructure file. You can provide

files that contain only the structures for particular reagents, or you can provide files that contain structures for a variety of reagents, from which CombiGlide will select out the specified types of reagents. In the pyrazole library example, you could supply one file containing all of the acid chlorides to be considered, one file for the hydrazines, and one for the isocyanates; or you could supply one large file containing structures for all three reagent types. The reagent files must be in either SD or Maestro format.

It is highly recommended that the reagent files contain a set of reagents that has been prefiltered to remove redundant compounds and any reagents that are not compatible with the synthetic approach. Prefiltering will reduce the amount of computational work needed to obtain a focused library and is more likely to lead to useful results.

If you do not already have a library of reagents, you can find links to sources of compounds at the HTScreening.net web site, http://www.htscreening.net/home, or the ZINC web site, http://blaster.docking.org/zinc. Most vendors supply structures of the compounds in 2D SD format.

You will also need to obtain a file containing the structure of the receptor. Protein structures can be obtained from the Protein Data Bank (PDB), http://www.rcsb.org/pdb. If you have the PDB database and Prime installed, and the environment variable `SCHRODINGER_PDB` is set, you can import a protein structure from the database directly into Maestro from the Import panel. See Section 3.1 of the *Maestro User Manual* for more information.

4.2 Choosing the Core-Containing Molecule

CombiGlide works by building side chains onto defined positions on a core-containing structure. You must therefore supply a structure that contains the core in addition to supplying the reagents. The reagent structures are used as a source of side chains. Even though in the actual chemical reactions, atoms from the reagents can be included into the core, these atoms are discarded when the side chains are added to the core.

There are some limitations on how you can select the reagents and attachment positions, which may affect the choices you make for the core-containing molecule.

- The bond that is replaced in the core-containing molecule and in the reagent when the side chain is added must be a single bond. In the pyrazole example in Figure 4.1, the bond that is replaced in the isocyanate is the single C–R' bond. The C=N bond in the isocyanate could not be chosen because it is a double bond, even though the corresponding bond in the core-containing molecule is a single bond.
- The bond must not be in a ring. The hydrazine N–N bond in the pyrazole example could not be chosen because it forms part of the pyrazole ring.

Core-containing molecules are also docked to define core poses, which are used to build and dock the substituted structures. By default, the minimally capped core molecule is used, but you can choose other molecules to dock for the core poses. When you choose core-containing molecules for use in defining core poses, it is often advantageous to select ones that are not so small that they can dock in poses that are unreasonable for the final substituted molecules. It does not matter how long or short the side chains are in the core-containing molecules, since they will be replaced. A good choice would be a known active.

Combinatorial Screening: Preparing Structures

When you have decided on the biological target and the synthetic route to the library you want to generate, and obtained structure files for the reagents and the receptor, you can proceed to the preparation of the structures for CombiGlide. Both the protein structure and the reagent structures must be properly prepared. You can perform these tasks in the Protein Preparation Wizard and Reagent Preparation panels. Following protein preparation, you must set up grids for Glide docking, which you can do in the Receptor Grid Generation panel.

5.1 Protein Preparation and Grid Generation

A prepared protein structure is required for CombiGlide docking. Protein preparation can include simplifying multimeric complexes, deleting unwanted waters or cofactors, correcting serious errors in the protein structure, adjusting bond orders, ionization states, and formal charges of the protein, metal ions, cofactors, and ligand, adding hydrogens, neutralizing appropriate amino acid chains, reorienting side-chain hydroxyl and thiol groups, and relieving steric clashes. The full procedure for protein preparation is described in Section 1.1 of the *Protein Preparation Guide*. The entire process can be performed in the Protein Preparation Wizard panel, which is described in Chapter 2 of the *Protein Preparation Guide*. You can open the Protein Preparation Wizard panel from the Workflows menu.

After running the protein preparation job, it is always advisable to check the structure for any remaining problems.

When you have prepared the protein, you must generate grids for docking with CombiGlide, using the Receptor Grid Generation panel. The grid generation process is described in detail in Chapter 4 of the *Glide User Manual*. You should refer to this chapter and follow the procedures in it to generate your grids. You can open the Receptor Grid Generation panel from the CombiGlide submenu of the Applications menu. This panel is also identical to the Glide version of the panel.

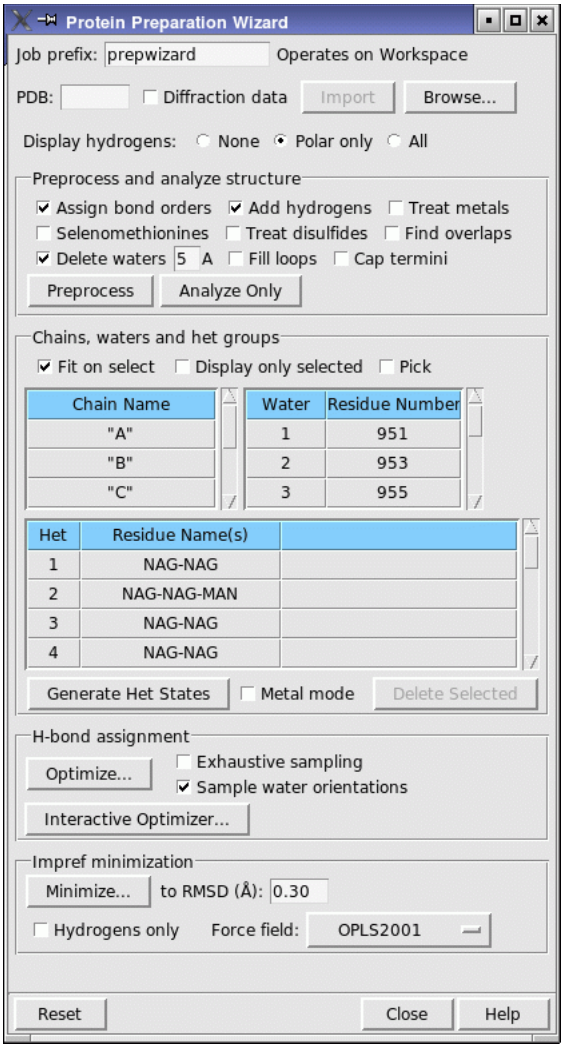


Figure 5.1. The Protein Preparation Wizard panel.

5.2 Reagent Preparation

Reagent preparation ensures that the input structures are all-atom, 3D structures, and that they have the appropriate information stored with them to construct the molecules that are docked or used for library enumeration. The main tasks in the reagent preparation process are to select the source of reagent structures, select a reagent type (a functional group), identify the bond in the functional group that is replaced when the reagent is added to the core, and perform 2D-to-

3D conversion, structure variation and cleanup. The first three of these tasks are performed in the Reagent Preparation panel, and options for the fourth are set up in this panel. A job is then run to obtain the structures from the input file and prepare them for use with CombiGlide.

To open the panel, choose Reagent Preparation from the CombiGlide submenu of the Applications menu. This panel contains three sections: one for selecting the input structure file, one for selecting the functional group, and one for setting options for structure conversion, variation, and cleanup.

You must run a reagent preparation job for each reagent type that will be used in combinatorial screening or in combinatorial library enumeration. A reagent type corresponds more or less to a functional group, plus information on which bond is to be replaced in the build process. The tasks described briefly above and in detail below must be performed for each reagent type that you plan to use. In the pyrazole example from Chapter 4, there are three reagent types: acid chlorides, hydrazines, and isocyanates.

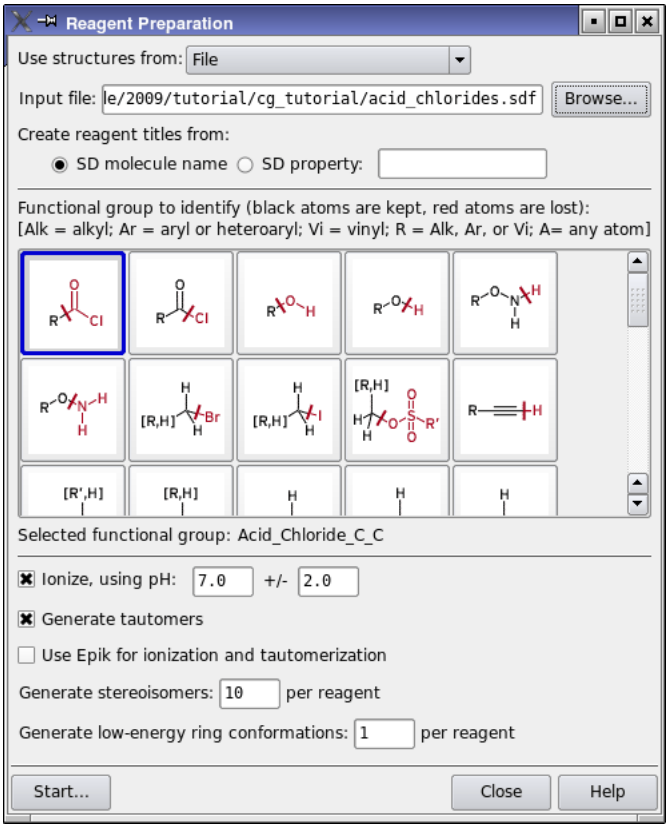


Figure 5.2. The Reagent Preparation panel.

A reagent preparation job is a special kind of LigPrep job that performs some tasks needed by CombiGlide in addition to the normal LigPrep tasks of ligand preparation and variation. For more information on LigPrep, see the *LigPrep User Manual*.

5.2.1 Choosing the Source of Reagent Structures

The first task is to choose the source of the reagent structures for a given reagent type. Because the reagent preparation job matches the selected functional group in each structure to a SMARTS pattern, structures that do not match are filtered out. The structures that match are written to a file and passed on to the next stage. The structure input file can therefore contain any structures, not just those that match the desired functional group. However, it is highly recommended that the input file you choose is prefiltered to eliminate redundant structures and structures that are not compatible with the synthetic approach.

There are some restrictions on the structures that you can use as input. If a structure contains more than one instance of the selected functional group, it will not be written to the output file unless it is symmetrical, in which case only one of the bonds will be designated as the bond to be broken in the process of building the library member from the reagents.

Once you have opened the Reagent Preparation panel, you can choose the source of structures from the Use structures from option menu. You can select structures from the Workspace, the Project Table, or from an external file. Structures from the Workspace or the Project Table are written to a Maestro file that is passed to the reagent preparation job.

If you choose File from the Use structures from option menu, you can enter the path to the file in the Input file text box, or click Browse to navigate to the file. The file must be in Maestro (`.mae`), SD (`.sd` or `.sdf`), or SMILES (`.smi`) format. The structures can be 2D or 3D, and are converted to all-atom 3D structures by the reagent preparation job.

If you selected an SD file, you can also specify the source of the reagent titles. Reagent titles are used for identification in CombiGlide, so it is important to have unique titles. If you do not make them unique, CombiGlide will do so for you. You can select SD molecule name to use the molecule name from the SD file, or select SD property and enter a property name in the text box to use the property for the reagent titles.

5.2.2 Selecting the Reagent Type

Selecting the reagent type involves choosing a functional group along with the bond that is to be replaced when the reagent is used to build structures for the library. A variety of predefined functional groups with corresponding bonds to be broken are displayed as a set of buttons in the center of the Reagent Preparation panel. The icon for each functional group marks the bond that is broken to attach the reagent to the core with a red line across the bond. The part of the reagent that is discarded is shown in red.

To select a reagent type, click the button that corresponds to the desired combination of a functional group and a bond to break. The selection is displayed below the scrolling region. The "short name" and the "long name" of the selected functional group are displayed below the icon area. Both names consist of a part that identifies the type of compound (acid chloride, primary amine, and so on) and a part that identifies the bond that is broken, separated by an underscore character. The part that identifies the bond consists of two element names separated by an underscore. The first name defines the atom that is kept when the reagent is added to the core, and the second name defines the atom that is discarded, along with anything attached to it. For example, Acid_Chloride_C_C is the long name for acid chlorides in which the carbon-carbon bond of the acid chloride group is broken. In this case the entire –COCl group is discarded, and the R group is added to the core.

A single class of compounds can have multiple icons, corresponding to different bonds that are broken. You should make sure that you have selected the icon with the bond to be broken that is the most appropriate for your molecular system. For example, acid chlorides can have the C–C bond or the C–Cl bond broken.

A detailed description of each functional group is provided in Table 5.1. The functional groups are defined using SMARTS patterns.

If the functional group that you want to use is not listed, you can add a custom functional group. Details on the requirements and procedure are given in Appendix A.

Table 5.1. Predefined functional groups with definitions

Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	Acid_Cl_C_C Acid_Chloride_C_C	R can be anything with a carbon attached to the carbonyl.
	Acid_Cl_C_Cl Acid_Chloride_C_Cl	See above
	Alc_C_O Alcohol_C_O	R can be an alkyl or aryl group. R cannot have a carbonyl attached to the oxygen of the alcohol.

Table 5.1. Predefined functional groups with definitions

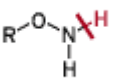
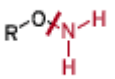
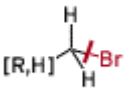
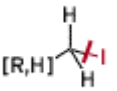
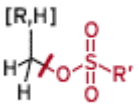
Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	Alc_O_H Alcohol_O_H	See above
	Alkoxylamine_N_H Alkoxylamine_N_H	R can be H or anything with a carbon attached to the oxygen of the alkoxylamine.
	Alkoxylamine_O_N Alkoxylamine_O_N	See above
	Alk_Br_C_Br Alkyl_Bromide_C_Br	R can be almost anything: H, alkyl, aryl, alkenyl, alkynyl; alkoxy, aryloxy, alkoxycarbonyl, or aryloxycarbonyl with oxygen attached to CH_2; silyl with silicon attached to CH_2; alkylamino, arylamino, alkylaminocarbonyl, or arylaminocarbonyl with nitrogen attached to CH_2; alkylthio, arylthio, alkylsulfinyl, arylsulfinyl, alkylsufonyl, or arylsulfonyl with sulfur attached to CH_2; ketone with carbonyl attached to CH_2; cyano. R cannot be chloro, iodo, chlorocarbonyl (carbonyl attached to CH_2).
	Alk_I_C_I Alkyl_Iodide_C_I	Same as for alkyl bromides
	Alk_Sulf_C_O Alkyl_Sulfonate_C_O	Same as for alkyl bromides
	Any_I_A_I Any_Iodide_A_I	A can be any atom for which force fields are available.

Table 5.1. Predefined functional groups with definitions

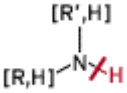
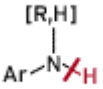
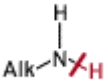
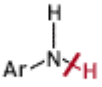
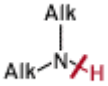
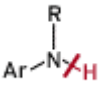
Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	Alkyne_C_H Alkyne_C_H	R can be H, alkyl, aryl, silyl.
	Amine_Gen_N_H Amine_General_N_H	R can be H, alkyl, aryl. R cannot have a carbonyl carbon attached to the nitrogen of the amine. R' can be H, alkyl, aryl. R' cannot have a carbonyl carbon attached to the nitrogen of the amine.
	Amine_Gen_Ar_N_H Amine_General_Aryl_N_H	Ar can be aryl. R can be H, alkyl, aryl. R cannot have a carbonyl attached to the nitrogen of the amine.
	Amine_Prim_Alk_N_H Amine_Primary_Alkyl_N_H	R can be H, alkyl. R cannot have a carbonyl attached to the nitrogen of the amine.
	Amine_Prim_Ar_N_H Amine_Primary_Aryl_N_H	Ar can be aryl.
	Amine_Prim_Gen_N_H Amine_Primary_General_N_H	R can be alkyl, aryl. R cannot have a carbonyl carbon attached to the nitrogen of the amine.
	Amine_Sec_Alk_N_H Amine_Secondary_Alkyl_N_H	R can be alkyl. R cannot have a carbonyl carbon attached to the nitrogen of the amine. R' can be alkyl. R' cannot have a carbonyl carbon attached to the nitrogen of the amine.
	Amine_Sec_Ar_N_H Amine_Secondary_Aryl_N_H	Ar can be aryl. R can be alkyl, aryl. R cannot have a carbonyl carbon attached to the nitrogen of the amine.

Table 5.1. Predefined functional groups with definitions

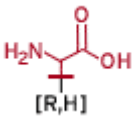
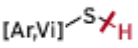
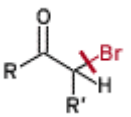
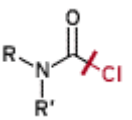
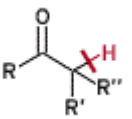
Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	Amine_Sec_Gen_N_H Amine_Secondary_General_N_H	R can be alkyl, aryl. R cannot have a carbonyl carbon attached to the nitrogen of the amine. R' can be alkyl, aryl. R' cannot have a carbonyl carbon attached to the nitrogen of the amine.
	A_Acid_C_C Amino_Acid_C_C	R can be H, alkyl, aryl.
	Ar_or_Vinyl_Br_C_Br Aryl_or_Vinyl_Bromide_C_Br	Ar is an aryl group; Vi is a vinyl (C=C) group. The aryl or vinyl group must be directly attached to the Br.
	Ar_or_Vinyl_I_C_I Aryl_or_Vinyl_Iodide_C_I	Ar is an aryl group; Vi is a vinyl (C=C) group. The aryl or vinyl group must be directly attached to the I.
	Ar_or_Vinyl_SH_S_H Aryl_or_Vinyl_Thiol_S_H	Ar is an aryl group; Vi is a vinyl (C=C) group. The aryl or vinyl group must be directly attached to the S.
	alpha_Br_CO_C_Br alphaBromocarbonyl_C_Br	R can be alkyl, aryl. R' can be H, alkyl, aryl.
	Carbam_Cl_C_Cl Carbamoyl_Chloride_C_Cl	R can be anything with a carbon attached to nitrogen. R' can be anything with a carbon attached to nitrogen.
	alphaCarbonyl_C_H alphaCarbonyl_C_H	R can be H, alkyl, aryl, alkoxy with oxygen attached to carbonyl. R' can be H, alkyl, aryl, carbonyl with carbonyl carbon attached to CH, cyano. R" can be H, alkyl, aryl, carbonyl with carbonyl carbon attached to CH, cyano.

Table 5.1. Predefined functional groups with definitions

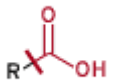
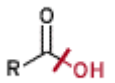
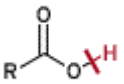
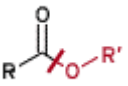
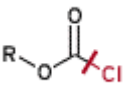
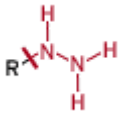
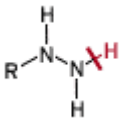
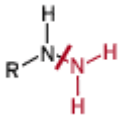
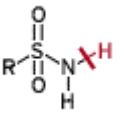
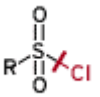
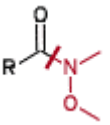
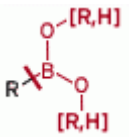
Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	C_Acid_C_C Carboxylic_Acid_C_C	R can be anything with carbon attached to carbonyl.
	C_Acid_C_O Carboxylic_Acid_C_O	See above
	C_Acid_O_H Carboxylic_Acid_O_H	See above
	C_Ester_C_O Carboxylic_Acid_Ester_C_O	R can be anything with carbon attached to carbonyl. R' can be anything with carbon attached to the oxygen except for a carbonyl carbon.
	Cl_Formate_C_Cl Chloroformate_C_Cl	R can be anything with carbon attached to oxygen.
	Hydrazine_C_N Hydrazine_C_N	R can be H or anything with a carbon attached to the nitrogen of the hydrazine.
	Hydrazine_N_H Hydrazine_N_H	See above
	Hydrazine_N_N Hydrazine_N_N	See above

Table 5.1. Predefined functional groups with definitions

Structure	Short name Long name	Definition of R, R', R", Alk, Ar, Vi, A
	Isocyanate_C_N Isocyanate_C_N	R can be anything with a carbon attached to the nitrogen of the NCO.
	Sulfonamide_N_H Sulfonamide_N_H	R can be anything with a carbon attached to the sulfur of the SO_2.
	Sulf_Cl_S_Cl Sulfonyl_Chloride_S_Cl	R can be anything with a carbon attached to the sulfur of the SO_2.
	Thiol_S_H Thiol_S_H	R can be H, alkyl, aryl, vinyl. R cannot have the carbon of a carbonyl attached to the sulfur.
	W_amide_C_N Weinreb_Amide_C_N	R can be alkyl, aryl. R cannot have a carbonyl carbon attached to the amide carbonyl.
	Boronic_C_B Boronic_Acid/Ester_C_B	R can be alkyl, aryl.
	Grignard_C_Mg Grignard_Reagent_C_Mg	R can be alkyl, aryl.

5.2.3 Setting Options for Structure Variation and Cleanup

In addition to defining the atoms of the reagents that will become the side chains, the reagent preparation process performs a 2D-to-3D conversion, generates all reasonable ionization and tautomeric states, and expands the stereochemistry. If the structures you have are already 3D structures in the appropriate ionization state, tautomeric and stereochemical form, you might

not need to generate variations. Many collections of reagents are provided in 2D form, with implicit hydrogen atoms. To generate 3D, all-atom structures for docking, some variation to obtain structures that represent the actual state under physiological conditions may be needed.

The lower portion of the Reagent Preparation panel allows you some control over the structure variation and cleanup process, by setting options for the following:

- Ionization state—You can request that the structure be protonated or deprotonated to represent the actual forms in water in the given pH range, by selecting Ionize, using pH and entering the pH value and range in the accompanying text boxes. If this option is not selected, the ionization state is left as is.
- Tautomerization state—You can request the generation of tautomers of the input structures by selecting Generate tautomers. Ionized structures might have a different tautomeric form from the un-ionized structures.
- Use of Epik for ionization and tautomerization—Epik is a more sophisticated tool for generating ionization and tautomerization states than the default tools provided with LigPrep, the `ionizer` and the `tautomerizer`. It is separately licensed, so you must have an Epik license to use this option.
- Stereochemistry—Stereoisomers of the input structures are generated in the absence of chirality information, either explicitly specified or deducible from the 3D structure. If the input structures are 2D and have chiral centers but do not have chirality information, you should ensure that the number specified in the Generate stereoisomers text box covers the possible combinations.
- Ring conformations—If the input structures contain rings that can exist in more than one low-energy conformation, enter the number of likely low-energy conformations in the Generate low-energy ring conformations text box. Ring conformations are not searched during docking, so the conformations must be set up during the reagent preparation process.

If you do not need to generate structural variations, you should deselect Ionize, using pH and Generate tautomers, and enter `1` in the Generate stereoisomers and Generate low-energy ring conformations text boxes.

5.2.4 Running the Reagent Preparation Job

When you have completed the tasks above, click Start. The Start dialog box opens, in which you can select job options and start the job. When you click Start in the Start dialog box, the Monitor panel opens, and displays information on the progress of the job.

The output structure file contains the prepared reagents. There might be several output structures per input structure if the molecules can exist in several low-energy forms. Consequently, there can be multiple structures for a given combination of reagents on a core (a "compound") and the final number of structures in the library can be greater than the final number of compounds.

The following output files are required for later use:

jobname`.bld`	Reagent structures in all modifications labeled for building
jobname`.log`	Overall log file for the job
jobname`.sqlite`	Sqlite database built from the `.bld` file
jobname`_vpost-stats.txt`	Text file giving summary statistics of molecules in the `.bld` file

5.2.5 Task Summary

To set up and run a reagent preparation job:

1. Choose the source of reagent structures
2. Select the functional group
3. Select 2D-to-3D conversion and structure variation (LigPrep) options
4. Click Start

5.3 Preparing the Core-Containing Molecules

The structures that you select for the core-containing molecules must be all-atom, minimized, 3D structures. If the structure you have does not meet these criteria, you can prepare it using LigPrep. For more information, see the *LigPrep User Manual*.

Note that only the ionization state and tautomer that you select is used for the core-containing molecule when you define the core in the Define Combinations step of the Combinatorial Screening panel or in the Combinatorial Library Enumeration panel. If you want libraries with multiple states of the core, you must generate the states of the core-containing molecule and select them one-at-a-time to generate a library based on each state.

Combinatorial Screening: The Process

The goal of combinatorial screening is to create a small combinatorial library of structures that are likely to have high activity towards to a selected target. The library is screened, first by docking to the target receptor and eliminating structures that do not dock well, then by analyzing the docked structures to define a small reagent set that is likely to have a large number of actives among the compounds that are generated.

Once you have decided on the chemistry, obtained files containing the reagents, the receptor and the core-containing molecule, and prepared the structures in these files, you can proceed with the combinatorial screening process. This process is managed from the Combinatorial Screening panel. To open the Combinatorial Screening panel, choose Combinatorial Screening from the CombiGlide submenu of the Applications menu in the main window.

6.1 The Combinatorial Screening Panel

The Combinatorial Screening panel is designed like a wizard, with five steps. To perform a combinatorial screening, you proceed through each step in turn. You do not have to complete all steps in a single session. The design allows you to exit at an intermediate stage and pick up the process again later at any step for which you have the required data.

Each set of data defined by proceeding through the steps in this panel is called a *run*. The results for each run are stored as a separate entity. You can open existing runs, and create and save runs from the File menu. Runs are saved in the Maestro project, so if you want to keep your results, you should create a named Maestro project to store them. See Section 8.1 of the *Maestro User Manual* for more information.

If you backtrack in a given run and make changes, the results for all the steps that depend on what you changed are discarded. When you make a change that affects later results, you are prompted to save the previous results and create a new run.

The step features occupy the center of the panel, and consist of a title with a brief description of the step at the top, a set of controls and tables for results, and a Back and a Next button at the bottom. The tasks for each step are summarized in the next section, and the steps are described in detail in the following chapters.

In addition to the step features, the panel contains a menu bar and an octagon button at the top, and a step guide just above the Close and Help buttons. These features are described below.

Table 6.1. Description of the File menu.

Menu Item	Description
New	Create a new run. Opens a dialog box to specify the run name. The new run becomes the current run.
Open	Open an existing run from the submenu. If there are more than 4 runs, choose More to open a dialog box and select a run.
Save As	Save the current run with a new name. Opens a dialog box to specify the new run name. The run is copied, and the renamed run becomes the current run.
Rename	Rename the current run. Opens a dialog box to specify the new run name.
Delete	Delete the current run.

The File menu

The File menu allows you to work with the runs that are available in the project. The items on this menu are described in Table 6.1.

The Step menu

The Step menu contains an item to display or hide the Guide, and items for each of the steps. The current step is marked with a red diamond. If the Guide is displayed, it is marked with a red square. The items for the steps that are not available are dimmed. You can go to any available step by choosing the corresponding menu item.

The job status button

When a job has been launched and is running, the icon on the job status button at the upper right of the panel turns green and spins. When the job finishes, the icon turns pink and stops spinning. If incorporation of results takes some time, the icon is replaced by an exclamation point while the incorporation is taking place. To monitor the job using the Monitor panel, click the button. For more information about monitoring jobs, see Chapter 3 of the *Job Control Guide*.

The Guide

The Guide displays the steps in the model as a set of buttons linked by lines. The buttons for the steps that are not available are dimmed. The current step is highlighted with a white background. You can go to any available step by clicking its button in the Guide. The Guide can be displayed or hidden from the Step menu.

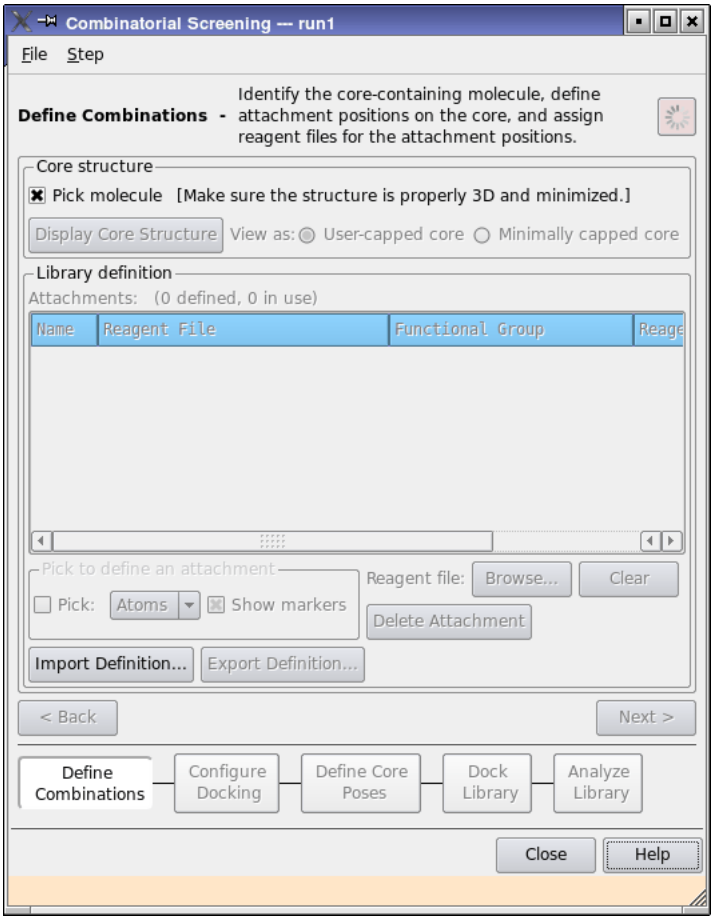


Figure 6.1. The Combinatorial Screening panel.

6.2 The Combinatorial Screening Process

The tasks involved in each step of the combinatorial screening process are summarized below.

The Define Combinations step

In this step, you first select the core-containing molecule that you want to use for generation of the library. The library is generated by replacing selected side chains in the molecule with other side chains from the reagent files. For each side chain, you identify the bond that defines where the substitution will take place (the *attachment position*), and specify a reagent file that contains the reagents with the side chains that you want to substitute. The bond is called the *grow bond*. The combination of an attachment position and a reagent file is called an *attach-*

ment. The core of the molecule is the part of the molecule that is left when you remove the side chains. The reagent libraries must be prepared beforehand, using the Reagent Preparation panel.

Once you have selected a molecule and the attachment positions for the reagents, and added the reagent files, you can proceed to the next step.

The Configure Docking step

The screening process uses CombiGlide docking to filter out reagents that produce molecules that do not score well. This step assumes that you have already generated the necessary Glide grid files, using the Receptor Grid Generation panel. In this step, you set options for the docking of the libraries to the receptor. As well as selecting a grid and making basic settings as for a Glide run, you can specify the full range of Glide constraints for the docked molecules.

After selecting a grid and setting any desired options, you can proceed to the next step.

The Define Core Poses step

In this step, you select the method for determining the poses of the core structure that you want to use in the docking step. These poses are generated by docking one or more molecules that contain the core. These molecules can be the core-containing molecule, the minimally capped core, or some other molecules that you choose. The *minimally capped core* is the core of the molecule with a minimal capping group at each attachment position. In most cases, the minimal capping group is defined by R, R', R"=Me or Ar=Ph for each reagent. See Table 5.1 on page 41 for definitions of R, R', and R".

You can constrain the core position to prevent it from moving to regions of the receptor that would generate unacceptable binding modes.

When you have selected a method, you can proceed to the next step.

The Dock Library step

In this step you run the docking jobs. The docking is performed in three stages. In the first stage, the core-containing molecules are docked to obtain a set of core poses, using whatever constraints on the core position you imposed.

In the second stage, each reagent from each reagent library is added in turn to the core, to produce molecules with a single side chain and the minimal capping groups at the remaining attachment positions. These molecules are docked using CombiGlide XP docking. Reagents that do not generate any good poses are eliminated. This stage can be run independently, and is known as the “single-position docking” stage.

In the third stage, the poses with a single side chain are combined intelligently to produce molecules that are likely to dock well. These molecules have side chains from the reagent libraries at all attachment points. This stage is the most time-consuming part. Glide constraints are only applied in this stage. This stage is known as the "combinatorial docking" stage.

After docking, you can export the results of either the single-position docking or the combinatorial docking for further examination. From this step you can also enumerate and dock the entire library without performing the initial screening.

Once you have docked the combinatorial library, you can proceed to the next step.

The Analyze Library step

In this step, the number of reagents used at each position is reduced to a small set by application of filtering and selection strategies. Filtering is based on structural properties such as molecular weight and counts of structural features, and ADME properties. These properties are generated by QikProp, which is run automatically if it is installed. The selection strategies identify the "best" reagents at each position, based on the GlideScore of each of the docked structures. When the small reagent set is selected, the library defined by this set can be enumerated.

Combinatorial Screening: Defining the Core and Its Attachments

The combinatorial screening process begins with the definition of the core and its attachments. The core-containing structure must be a 3D, all-atom, minimized structure. Likewise, the reagents used for the attachments must be 3D, all-atom, minimized structures, and also include information that defines the side chain to be attached to the core. Preparation of the core-containing molecule and the reagents is described in Chapter 5. The task of selecting the core-containing molecule, defining the attachment points, and associating a set of reagents with each attachment point is carried out in the Define Combinations step of the Combinatorial Screening panel.

7.1 Selecting the Core-Containing Molecule

The first task in this step is to select the core-containing molecule. When you first enter the Define Combinations step, Pick molecule is selected in the Core structure section, and most other controls are unavailable (see Figure 7.1).

To select the core-containing molecule:

1. Ensure that the desired molecule is displayed in the Workspace.

   You can import the structure if necessary, or open the Project Table panel to display the structure, while the Combinatorial Screening panel is open. If you import the structure, ensure that Replace Workspace is selected.

2. Click on an atom in the core-containing molecule.

   Any structures other than the core-containing molecule are undisplayed. The other controls in the step become available.

The molecule that you picked is copied into the CombiGlide run. Any changes you make to the structure of the molecule are not saved in the run. If you make changes to the structure, you must pick the molecule again to make it the core-containing molecule. If you have already picked a molecule, picking a new molecule deletes all information about attachments.

After you pick the core-containing molecule, Pick molecule is deselected, and most other controls become available. The Pick option is selected in the Pick to define an attachment section. The core-containing molecule is now defined, and you can define the positions of the attachments, which also defines the actual core.

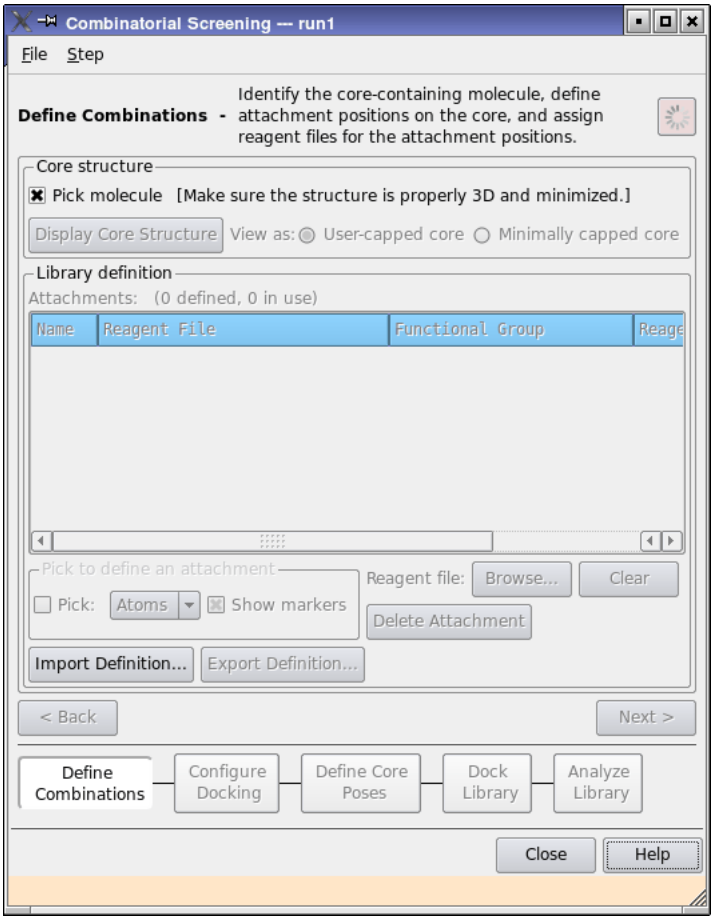


Figure 7.1. Initial view of the Define Combinations step.

7.2 Defining Attachments

Once you have a core-containing structure, you can define the attachments. An *attachment* is defined as the combination of an attachment position and a file that contains the reagents that will be attached at this position. The attachment position is defined by the atom in the core to which a side chain is attached and the atom in the side chain that is bonded to it. To make the definition, you must select both the atom that is kept in the core, and the atom that is bonded to it in the side chain. The order in which you pick these atoms determines which is the core and which is the side chain. The bond so defined is called the *grow bond:* it is essentially a vector that points from the core to the side chain. The direction is the direction in which the side chain is "grown" from the core.

Having defined the attachment position and the side chain, the next task is to select a set of reagents for this attachment position from which the side chains are taken. The reagents are contained in files that you prepared using the Reagent Preparation panel.

To define an attachment:

1. Pick two atoms in the core-containing molecule that define the attachment position, the atom that is kept first, then the atom that is deleted.

   Alternatively, choose Bond from the Pick option menu in the Pick to define an attachment section, and click on the appropriate bond. When you click, make sure you click on the end of the bond closest to the side chain, to ensure that the grow bond points in the right direction. This is especially important for the first attachment. For subsequent attachments, a warning is posted if the grow bond points in the wrong direction.

   After picking the second atom (or the bond), a gold arrow is displayed over the bond, and the Select Reagent File dialog box opens.

2. Select the appropriate reagent file, and click OK.

   The reagent files have a `.bld` extension: they are Maestro-format files that have special information for the Maestro builder, which does the library enumeration. The dialog box closes, and the information in the row of the Attachments table is completed. The size of the library with the current set of attachments is given in the lower portion of the panel.

   The arrowhead is replaced with a gold sphere centered on the side chain atom.

If you choose the wrong location or the wrong direction for an attachment, you can delete it by selecting it in the table and clicking Delete Attachment. When the error is recognized by CombiGlide, such as if you pick a bond in a ring, a warning is posted and the attachment is cleared. If you simply attached the wrong file, you can select the attachment in the table, and click the Browse button below the table to select the correct file, or click Clear to remove the attachment file.

For each attachment a minimal capping group is automatically defined. This minimal capping group is usually a small instance of the side chain in which R, R', R"=Me or Ar=Ph for the chosen reagent type. Adding the minimal capping group to the core defines the *minimally capped core*. You can view the minimally capped core by selecting Minimally capped core from the View as options in the Core structure section. To redisplay the original core-containing structure, select User-capped core.

Information for the attachments that have been defined is listed in the Attachments table. The table columns are described in Table 7.1. You can select multiple rows in the table with the usual shift-click and control-click actions. The markers for the selected attachments are colored turquoise in the Workspace.

Table 7.1. Description of Attachments table columns

Column	Description
Name	Label for the position of the attachment. Default is 1, 2, 3, etc. Editable.
Reagent File	Name of the file containing the reagents, minus the extension. The tooltip for this table cell displays the full path to the file.
Functional Group	Long name of the functional group that describes the reagents.
# Reagents	Number of reagents in the file.

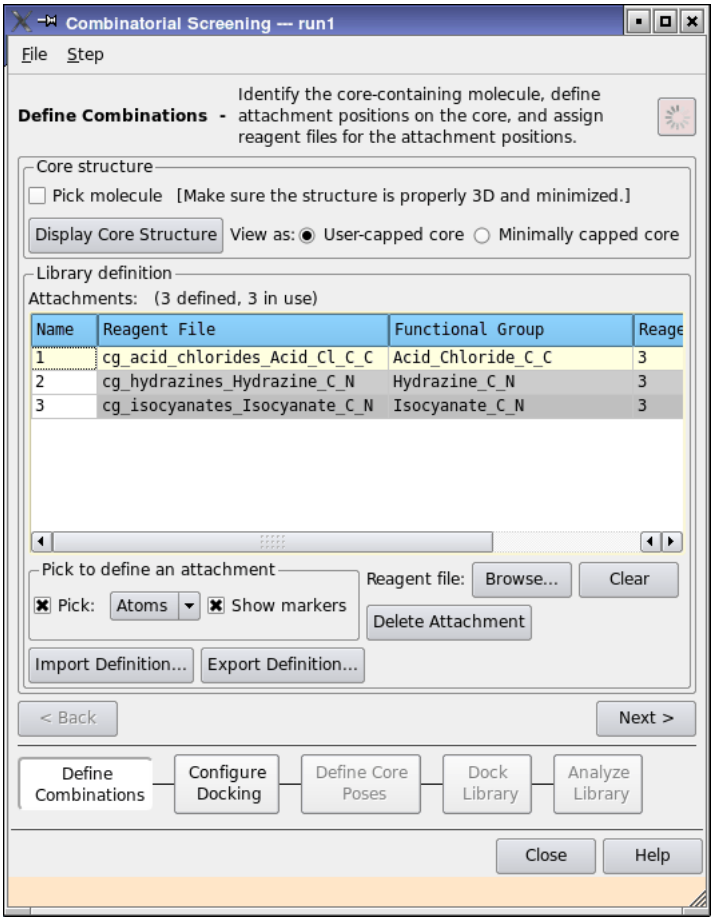


Figure 7.2. The Define Combinations step after defining attachments.

Right-clicking in a table row displays a menu with four items: Select Reagent File, Clear Reagent File, Reagent File Statistics, and Delete Attachment. The actions are applied to all selected rows. The Reagent File Statistics item displays information on the selected reagent files from the reagent preparation process, including the number of structures that were generated at each step and molecular weight data statistics. The other items perform the same actions as the Browse, Clear, and Delete Attachment buttons.

Note: If you define only one attachment point, the only workflow available to you is fully enumerated docking. The postdocking library analysis depends on the existence of more than one attachment point.

7.3 Saving and Loading Library Definitions

The combination of a core structure and its attachments defines a combinatorial library. You can save this definition for future use, by clicking Export Definition. A file browser opens, in which you can navigate to the desired location and specify a name. The library definition is saved as a gzipped tar file, and the extension `-comdef.tar.gz` is added to the name automatically. The library definition includes the core structure, which is written to the file `core-comdef.mae`, and the files for each of the attachments. These files are all added to the tar file.

If you want to use a previously-defined library definition, including the core structure and the attachments, click Import Definition, and navigate to the desired gzipped tar file. The default filter for the file selector is `-comdef.tar.gz`, so you should only see valid library definitions.

Combinatorial Screening: Setting Up for Docking

The center of the combinatorial screening process is the docking of the library members using CombiGlide XP docking. After you have defined the library, the next task is to set up the parameters and options for docking.

The docking of the library is performed in three stages:

- **Core docking.** In this stage, the core-containing molecules are docked to determine the best core poses. By default, the user-capped core is used. These poses are used as the initial poses for the substituted structures that constitute the library members. Two jobs are run in this stage, called `core` and `mcore` docking.
- **Single-position docking.** In this stage, the side chains at each position are added to the core, one at a time, to generate a set of structures with side-chain substitution at a single position. The other attachment positions are capped with either the minimal capping group or the user-defined capping group for that position. These singly-substituted structures are then docked, and passed through a selection process that eliminates reagents that are likely to result in poor binding.
- **Combinatorial docking.** The final stage is to dock a selected number of fully substituted structures that were predicted to have the highest probability of binding to the receptor by the selection algorithm.

Setting up the docking calculations is done in two steps: a configuration step, in which Glide parameters and constraints are set, and a core pose definition step, in which the structures to use for the core docking are selected and any constraints on the core position are set. These tasks are performed in the Configure Docking and Define Core Poses steps of the Combinatorial Screening panel.

8.1 Making Glide Settings

The Glide settings are made in the Settings tab of the Configure Docking step. This tab is a combination of parts of the Settings tab and the Ligand tab from the Glide Ligand Docking panel. Because CombiGlide uses Glide XP docking and the ligands are predefined, only the relevant options from these tabs are available. There are three sections in this tab: Receptor grid, Docking, and Van der Waals radii scaling.

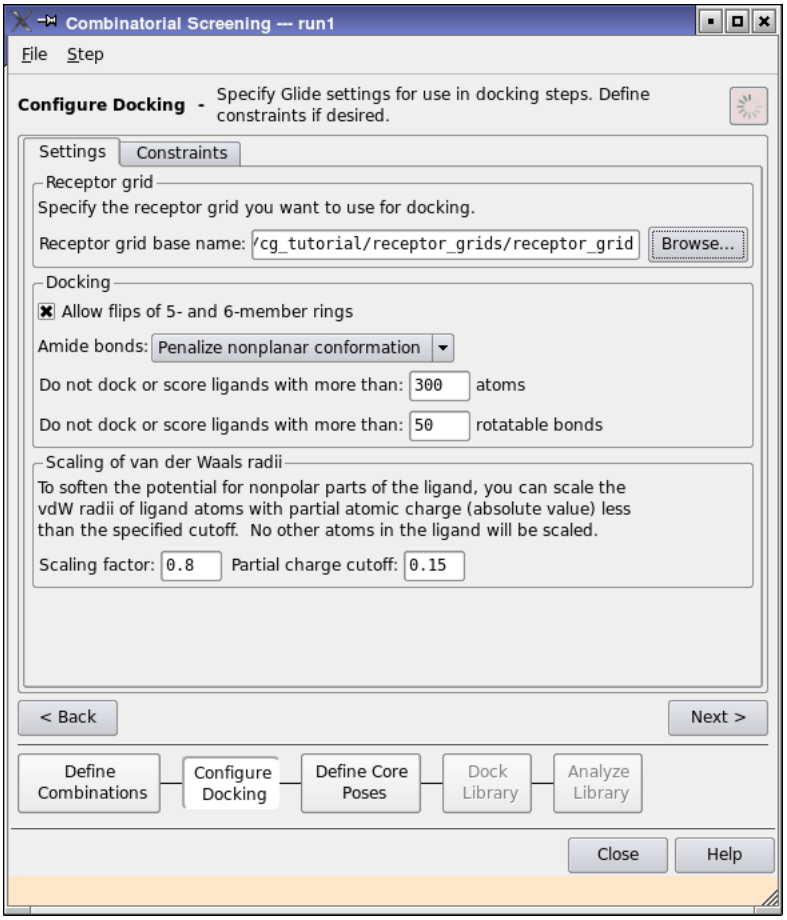


Figure 8.1. The Settings tab of the Configure Docking step.

The first task is to select the receptor grid. You can either enter the path to the receptor grid file in the Receptor grid base name text box, or click Browse and navigate to the receptor grid file.

If you have not generated a grid, you must do so before proceeding, in the Receptor Grid Generation panel. You can open this panel from the CombiGlide submenu of the Applications menu in the main window. When you have generated a grid, you can reopen the Combinatorial Screening panel and return to this step.

Next, you can set options and parameters as listed below. The ring flip and amide bond rotation options are applied in all docking stages. The limits on the maximum number of atoms and rotatable bonds are only applied in the combinatorial docking stage. Structures exceeding these limits are not built and thus are not docked.

For more information on these options, see Chapter 5 of the *Glide User Manual*.

Allow flips of 5- and 6-member rings

Select this option to allow the conformation of nonaromatic 5- and 6-membered side-chain rings to be varied during docking. This option is selected by default.

Amide bonds

Choose an option from this option menu for the treatment of amide C–N bonds. The options are Penalize non-planar conformation, which penalizes amide bonds that are not cis or trans when scoring the docked molecules, Vary amide bond conformation, which allows nonplanar amide bonds, Retain original amide bond conformation, which freezes amide bonds in their input conformation throughout docking, and Penalize cis amides, which penalizes cis amide bonds when scoring the docked molecules. The default is to penalize nonplanar amide bonds.

Do not dock ligands with more than *n* atoms

Enter a number to restrict the size of the molecules that are docked. The default is 120. This restriction is only applied in the combinatorial docking stage.

Do not dock ligands with more than *n* rotatable bonds

Enter a number to restrict the number of rotatable bonds to sample during docking. The default is 20. This restriction is only applied in the combinatorial docking stage.

Van der Waals radii scaling

Enter a scaling factor and a partial charge in the text boxes to scale the van der Waals radii of atoms with small partial charges. The scaling simulates the flexibility of nonpolar groups.

8.2 Setting Glide Constraints

Constraints can also be used with CombiGlide and are specified in the same manner as with Glide. With CombiGlide, constraints are only turned on during the combinatorial docking stage, i.e. docking of the fully substituted structures. This prevents the docking algorithm from forcing the side chains at different positions on the core into the same binding pocket during the single-position docking. In the combinatorial docking stage, constraints can be used to filter out reagents whose side chains cannot satisfy the relevant constraints.

The Constraints tab of the Configure Docking step tab contains controls for applying Glide constraints to the docking of the combinatorial library. The contents of this tab are identical to those of the Constraints tab in the Glide Ligand Docking panel. For information on setting constraints, see Section 5.5 of the *Glide User Manual*.

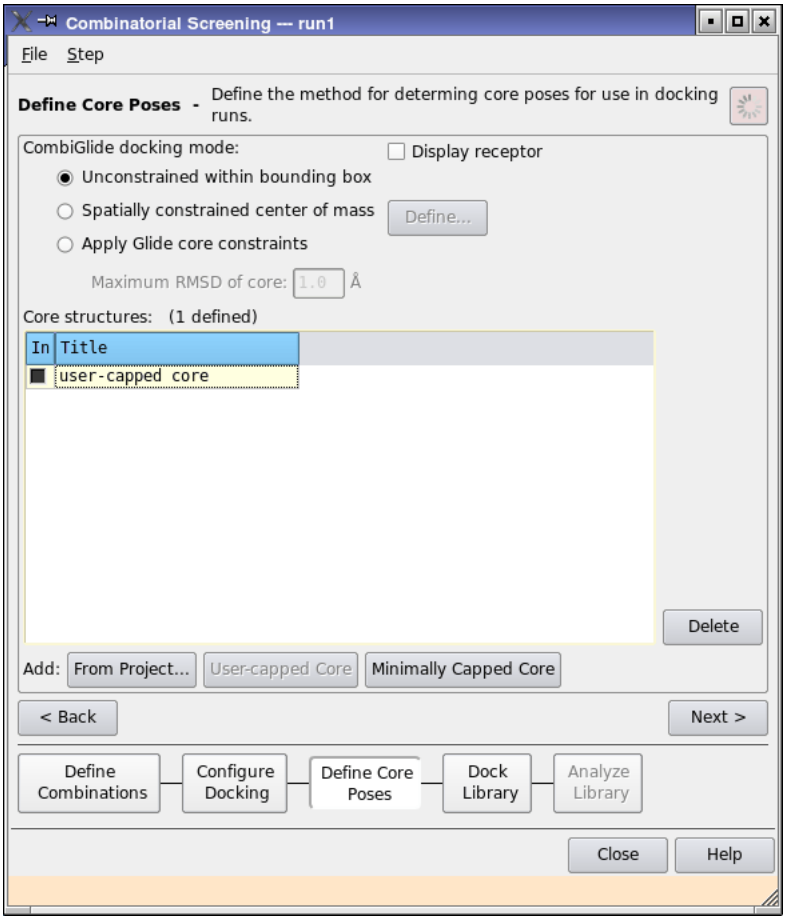


Figure 8.2. The Define Core Poses step.

8.3 Defining the Core Poses

The final setup task is to determine how the core poses will be generated in the first stage of the docking step. The core poses can be generated with any molecule that includes the core, not just the structure you selected to define the core in the Define Combinations step. This task is carried out in the Define Core Poses step, and involves selecting a docking mode and specifying the structures that will be docked.

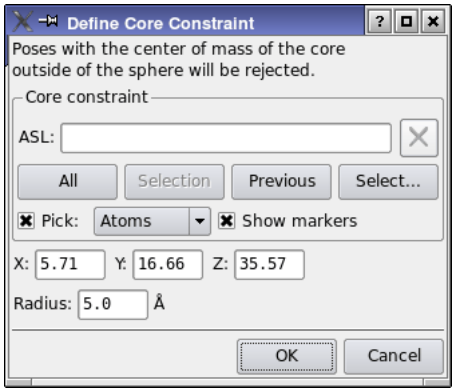


Figure 8.3. The Define Core Constraint dialog box.

8.3.1 Selecting a Docking Mode

The first stage in the docking step is to dock the core-containing molecules. This stage does not apply any Glide constraints that you have set. If the molecules used to define the core poses are smaller than the original core-containing molecule, they might bind to locations in the active site that are not accessible to the original. Many of these poses could be undesirable, for example if the core migrated to a location in which the fully substituted molecules could not dock. CombiGlide therefore supplies three docking modes, described below, two of which allow some kinds of constraints to be applied.

Unconstrained within bounding box

This is the default option and permits the molecule the widest range of movement within the grid bounding box during docking. If the molecules you select for the core poses are likely to dock in reasonable poses, select this option.

Spatially constrained center of mass

With this option, the core center of mass is constrained during docking to a sphere of a given radius. Constraining the center of mass allows the core-containing structures to rotate but prevents them from moving to another possible binding site.

The position of the sphere is defined by picking atoms in the Workspace, and you can choose atoms in the ligand or the receptor, or both. If you want to display the receptor in order to pick atoms, click Display Receptor. You can also display any of the core-containing structures by clicking the appropriate In column of the Core structures table. However, you should make sure that these structures are aligned to the receptor, otherwise picking atoms from them will be of no value. When you have the desired structure in the Workspace click Define. In the

Define Core Constraint dialog box, use the picking tools to pick the atoms whose centroid defines the sphere center.

The radius of the sphere is 1.0 Å by default, and can be set in the Define Core Constraint dialog box. The sphere radius must not extend beyond the enclosing box of the grid. You can also set the coordinates of the center of mass manually.

Apply Glide core constraints

This option allows you to define the core poses by the positions of the core-containing structures, and provides the greatest degree of constraint on the core poses. These structures must be already aligned to the receptor. When these structures are docked, poses for which the RMSD of the core atoms exceeds the specified threshold are rejected. One way of obtaining these core-containing molecules is to dock molecules that contain the core onto the same receptor using Glide, with whatever constraints you want to apply. Another is to use the crystal structure of the chosen receptor complexed with a core-containing molecule.

8.3.2 Specifying Structures for the Core Poses

The structures that will be docked to define the core poses are listed in the Core Structures table. You can add structures to the table with the Add buttons below the table, and you can delete structures from the table by selecting them and clicking Delete.

By default, only the user-capped core is listed. This is the core-containing molecule that you selected in the first step. If you delete this molecule, you can add it again by clicking User-capped Core. You can also add the minimally capped core to the table by clicking Minimally Capped Core. To add structures from the Project Table, click From Project, and select the molecules in the entry chooser that opens. The entries must satisfy the following criteria:

- They must contain only a single molecule
- The molecule must contain the core structure

No checking is done for duplicates, so you must ensure that you don't add the same structure more than once. The structures are copied into the run, so that any changes in the Project Table do not affect these structures.

If a crystal structure or docked pose is used for the core structure, the associated protein structure must be superimposed on the receptor structure from which the grids were calculated to ensure that the core poses are in the correct frame of reference. You can superimpose the protein structures in the Superposition panel, which you open from the Tools menu in the main window.

Combinatorial Screening: Docking the Library

The docking step is the heart of the combinatorial screening process, where the docking jobs are run to screen out molecules that do not dock well and thereby eliminate reagents. This step is carried out in the Dock Library step of the Combinatorial Screening panel.

First, the core-containing structures are docked. The resulting poses are clustered, and poses from each cluster are selected until the maximum number is reached. This ensures the widest diversity of core poses while limiting their number. The core poses are used as the initial positions of the substituted molecules in the subsequent docking stages.

The screening is performed in two stages. In the first stage, structures that have only a single substitution are docked. These structures are generated by adding the reagents, one at a time, to the minimally capped core or the user-capped core. The result is a set of molecules (a "single-position library") for each attachment position with a side chain at that position and the minimal capping group at the other positions. The side chains added are taken from the expanded set of reagents, in which different ionization states, stereoisomers, conformers, or tautomers can be represented. In the pyrazole example, there are three attachment positions and thus, three single-position libraries will be generated and docked. This portion of the virtual library evaluation is an additive process: the number of dockings required for this step is the sum of the number of side chains for each position.

Once the single-position docking results are returned, a selection algorithm determines which fully substituted cores have the highest probability of binding to the receptor. Structures from the single-position docking stage that do not dock well are eliminated. Reagents for which no structures docked well are removed from the reagent list for the appropriate attachment position. The selection algorithm takes into consideration the XP GlideScores of the docked poses from the single-position docking. It also checks that the side chains it has identified for a particular fully substituted core are not predicted to compete for the same part of the receptor binding site.

Finally, the structures that are identified by the selection algorithm are generated and docked. This set of structures is not a combinatorial library, in which each side chain occurs at each position.

As an alternative, the single-position docking and the selection procedure can be bypassed, and the entire library can be enumerated and docked.

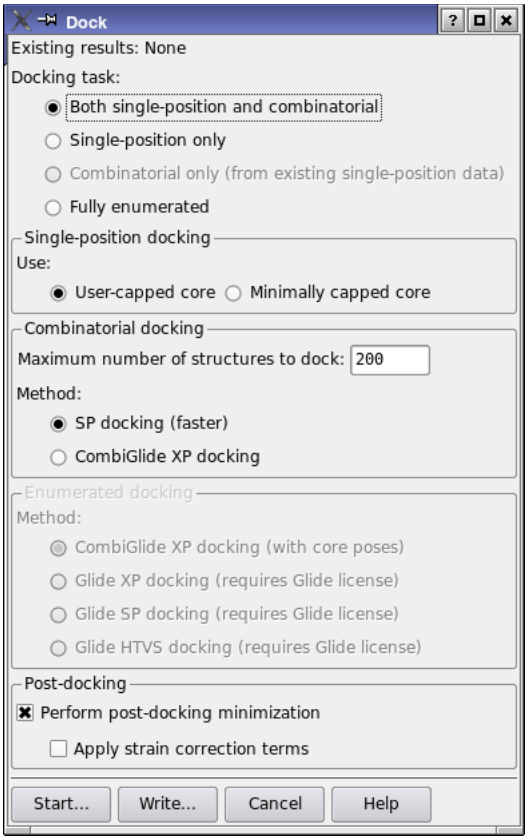


Figure 9.1. The Dock dialog box.

9.1 Running the Docking Jobs

The docking jobs are set up and run from the Dock dialog box, which opens when you click Dock. This dialog box offers choices for the docking task and options for each of these tasks.

If you want to continue on in the workflow to analyze the docking results and select the optimal library, you need to run both the single-position docking and the combinatorial docking stages. These two stages can be run in the same job or separately. If you want to examine the results of the single-position docking before continuing with the combinatorial docking, select Single Position only. When you want to proceed to the second stage, open the Dock dialog box again, and select Combinatorial only. The results of the first stage are then used to run the combinatorial docking. To run both stages in the same job, select Both single-position and combinatorial. The core docking is run automatically before the single-position docking.

The status of any existing results is displayed at the top of the dialog box, to help you choose the appropriate task.

Once you have chosen the task, you can choose options for the two docking stages. For single-position docking, you can use either the user-capped core or the minimally capped core as the molecule on which the single substitutions are made. The benefit of the user-capped core is that the side chains on the unsubstituted positions are usually larger and may prevent docking in unreasonable poses. Using the minimally capped core allows greater flexibility in docking poses, and might identify poses that would otherwise be missed.

For the combinatorial docking, you can set a limit on the number of fully substituted structures to dock in the Maximum number of structures to dock text box. The default is 20 times the sum of the number of side chains at each position. If you plan to use Glide constraints, you should significantly increase the limit to ensure sufficient output structures. You can also choose whether to use SP docking or CombiGlide XP docking.

If you want to dock the entire combinatorial library, select Fully enumerated for the docking task. You can then select the Glide docking mode. In addition to using CombiGlide XP docking, you can select any of the Glide docking modes: XP, SP, or HTVS. You must have the appropriate Glide license for Glide docking. If you have a single attachment point, this is the only option that is available.

For any of the tasks, you can also choose to perform a post-docking minimization and adjust the scoring for ligand strain, in the Post-docking section. Post-docking minimization usually results in better poses, because it adjusts bond lengths and angles as well as dihedrals. The ligands are rescored after the post-docking minimization, with the strain correction if it included. For more information, see Section 5.7.2 of the *Glide User Manual*.

When you have chosen all the options, click Start, set any job options in the Start dialog box, such as selection of a host and distribution of the job over multiple processors. Click Start to submit the job.

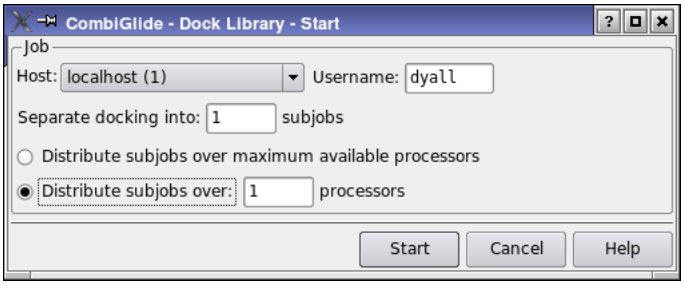


Figure 9.2. The Start dialog box.

When the combinatorial docking stage finishes, if QikProp is available, it is run for the completed structures, and the QikProp properties are added to the run. These properties can be used in the Analyze Library step.

To cancel the current docking job, click Cancel, then click OK in the confirmation dialog box. If you are running the combinatorial docking job, and the single-position docking job has already finished, the results of the single-position docking job are retained.

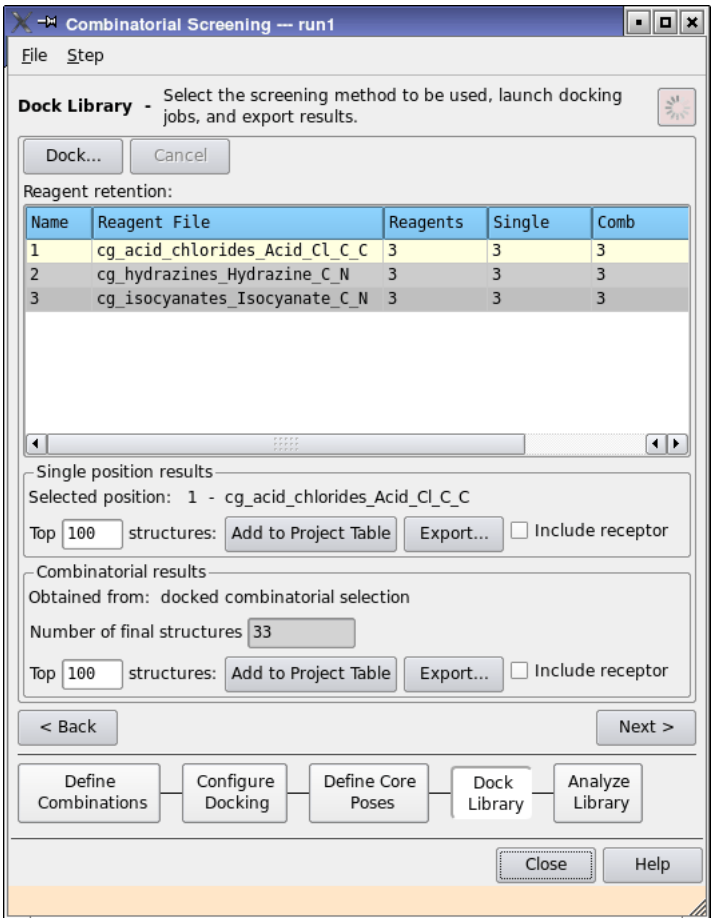


Figure 9.3. The Dock Library step.

9.2 Docking Results

Statistics on the reagents that survive the various docking stages are added to the Reagent retention table when the docking jobs are finished. The table is noneditable, and its columns are described in Table 9.1. The number of structures actually generated is reported in the Combinatorial results section. This number can be less than the product of the numbers in the Comb column of the Reagent retention table, because of the limits imposed and because not all structures generated from the single-position docking necessarily dock well.

Table 9.1. Description of Reagent retention table.

Column	Description
Name	Attachment position label
Reagent File	Name (minus extension) of the file that contains the reagents for the specified position.
Reagents	Number of reagents in the original reagent file.
Single	Number of reagents from this file that survived the single-position docking.
Comb	Number of reagents from this file that survived the combinatorial docking. This number includes all reagents that contributed to any structure that docked well.

Once the results are available, you can add them to the Project Table or export them to a file. The structures from both stages are sorted by their GlideScores. Because there may be variations of each reagent (such as different ionization states, tautomers, conformers), it is the top-scoring variation for each reagent that is exported. For the single-position docking results, multiple poses are included in the set that is exported: thus the top-scoring variation for each pose for each reagent is exported. The set of singly substituted or the set of fully substituted structures is added to the Project Table as an entry group, with QikProp properties if these are available. If the receptor is included, it is the first entry in the entry group or the file.

To export a set of structures from single-position docking:

1. Select the row in the Reagent retention table for the desired reagent type.
2. Enter the number of reagents in the Top *n* structures text box in the Single position results section.
3. (Optional) If you want to include the receptor, select Include receptor.
4. Click Add to Project Table or Export.
5. If you are exporting to a file, navigate to the directory, enter the file name, and click Export.

To export a set of structures from combinatorial docking:

1. Enter the number of structures in the Top *n* structures text box of the Combinatorial results section.
2. (Optional) If you want to include the receptor, select Include receptor.
3. Click Add to Project Table or Export.
4. If you are exporting to a file, navigate to the directory, enter the file name, and click Export.

Combinatorial Screening: Focusing the Library

The final step in the combinatorial screening process is to focus the library to a small number of compounds generated from a few reagents at each position. This focused library should contain a large proportion of compounds that are likely to bind well. The focusing is carried out by applying various selection strategies, by filtering on molecular and predicted ADME properties, by manual selection, and by setting limits on the number of reagents at each position and on the overall library size. The purpose of this step is to select sets of optimal reagents for each attachment position that are small enough and chosen well enough for practical synthesis and screening of the resulting real combinatorial library.

The selection and filtering can be repeated as many times as you like, and the results of each selection and filtering job can be stored and compared. You can combine selection, filtering and manual inclusion or exclusion of reagents. When you are satisfied with a selection, you can run a job to create and save the structures in the library.

The focusing of the library is driven from the Analyze Library step of the Combinatorial Screening panel. The selection and filtering tasks are set up and run from the Filter and Select dialog box, which you open by clicking Filter and Select. This dialog box is nonmodal, so you can interact with the Workspace and the Combinatorial Screening panel while it is open.

10.1 Selection Strategies

CombiGlide has three automated selection strategies, which can be supplemented with manual selection. The automated strategies use the GlideScore to select the reagents that produce the greatest number of structures that have a good GlideScore.

The selection strategies work by filling lists of reagents at each position, until the limits on the minimum and maximum number of reagents and the overall library size are met. The molecules from the docking step are ordered according to their GlideScore or some derived quantity. Each molecule is considered in turn, and the reagents that were used to build that molecule are added to the corresponding lists.

The strategies are selected in the Selection Strategy tab of the Filter and Select dialog box.

The first strategy selects reagents based on the single-position docking. In this strategy, the lists for each position can be filled independently, from the docking results for each position. This is the only strategy available if you do not have combinatorial docking results. To apply this strategy, select Single position under Selection strategy, and click Apply or OK.

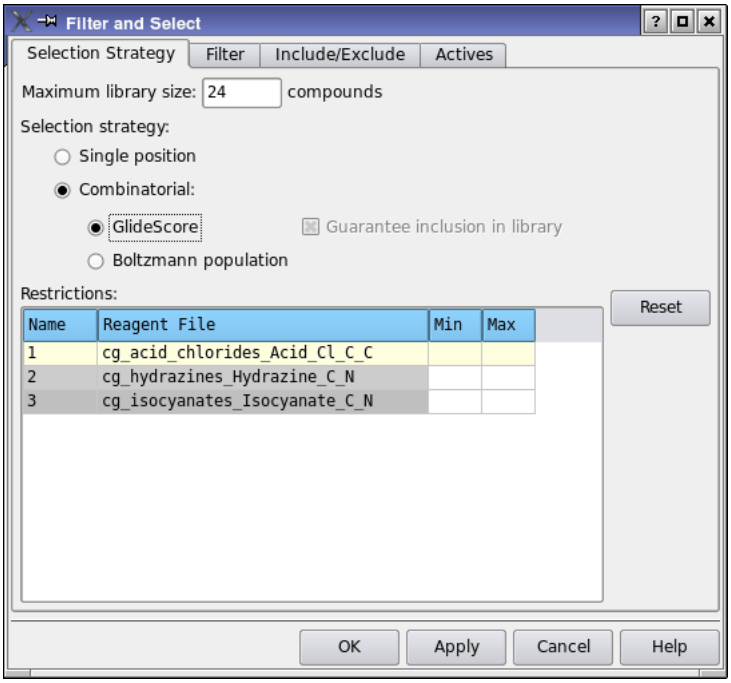


Figure 10.1. The Selection Strategy tab of the Filter and Select dialog box.

The second strategy selects reagents based on the GlideScores from the combinatorial docking. Starting from the structure with the best GlideScore, the reagent for each position is added to the list for that position. The reagents for the next best structure are then added to the lists, unless they have already been added, and so on, until the conditions on the minimum and maximum reagents at each position and on the maximum library size are met. When a list at a particular position is filled, there are two options for adding structures to the remaining partially filled lists. The first option is to add reagents from only those molecules that have reagents at the filled position that are in the filled list. The second option is to add reagents from any molecule, regardless of whether the reagent at the filled position is in the filled list. The first option ensures that all molecules used to determine the library are in the library. The second option ensures that the reagents came from the structures with the best GlideScores. To apply this strategy, select Combinatorial under Selection strategy, then select GlideScore. If you want to ensure that the library contains the molecules used to determine it, select Guarantee inclusion in library. This option is only available if you have minimum and maximum numbers in the Restrictions table (see below). Click Apply or OK to start the selection job.

The third strategy uses the GlideScore from the combinatorial docking as a free energy, and calculates the normalized Boltzmann population for each docked molecule. The populations are summed for each reagent, and the reagent populations are converted to a relative

GlideScore using a free-energy relation. Reagents are added to the lists in order of relative GlideScore, from lowest to highest. This strategy samples the most frequently occurring reagents, weighted according to GlideScore: reagents coming from molecules with good GlideScores are weighted more heavily than those with poor GlideScores. For each position, the list of reagents is filled, starting from the reagent with the lowest relative GlideScore, until the conditions on the minimum and maximum reagents at each position and on the maximum library size are met. To apply this strategy, select Combinatorial under Selection strategy, then select Boltzmann population. Click Apply or OK to start the selection job.

In addition to applying a selection strategy, you can set the minimum and maximum number of reagents at each position, and set an overall library size. The maximum library size can be entered in the Maximum library size text box. The minimum and maximum number of reagents at each position can be set by editing the table cells in the Restrictions table, which is described in Table 10.1.

Table 10.1. Description of the Restrictions table.

Column	Description
Name	Attachment position label
Reagent File	Name of reagents file for this attachment position, without the extension.
Min	Minimum number of reagents to include at this position. Editable.
Max	Maximum number of reagents to include at this position. Editable.

10.2 Manual Selection of Reagents

You can manually select reagents that must be included in the library or that must be excluded from the library in the Include/Exclude tab of the Filter and Select dialog box. This allows you to enforce the selection of reagents that did not score well, or exclude reagents that did score well, based on your knowledge of the chemistry, or to exclude reagents based on their availability.

To select reagents for inclusion or exclusion:

1. Select an attachment position from the Position option menu.

   All the reagents in the reagent file for that position are displayed in the Reagents list.

2. (Optional) Filter the reagents by entering a string in the Filter text box.

   You can use the "*" character to represent an arbitrary string. All reagents whose titles contain the specified string are listed in the Reagents list when you press ENTER.

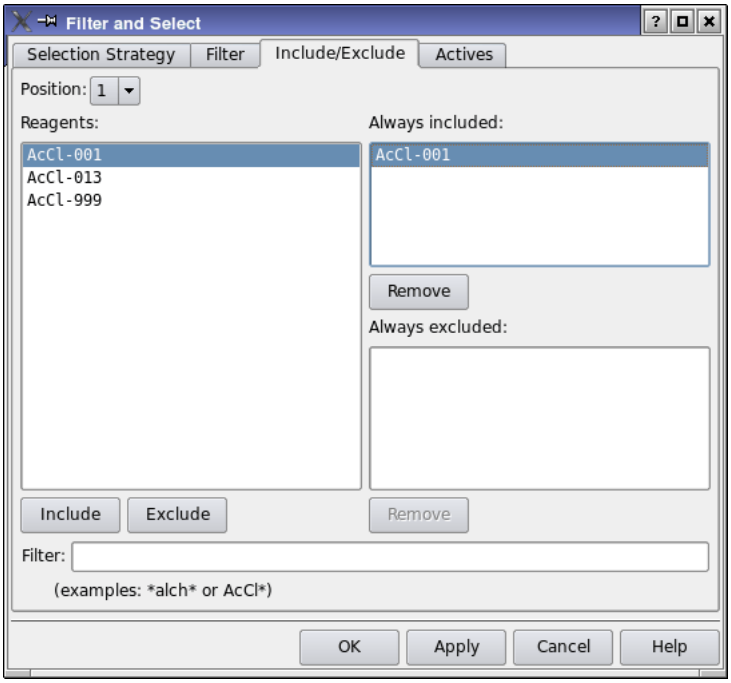


Figure 10.2. The Include/Exclude tab of the Filter and Select dialog box.

3. Select reagents from the Reagents list.
4. Click Include to add the selected reagents to the Always Included list, or click Exclude to add the selected reagents to the Always Excluded list.

If you add a reagent to one list that is already in another list, it is removed from the other list.

To remove reagents from a list:

1. Select the reagents in the list.
2. Click the Remove button immediately below the list.

You can include or exclude reagents directly from the Analyze Library step, by right-clicking a reagent in the Reagents at position *N* table. When you do so, the Filter and Select dialog box opens so that you can re-run the library selection job with the newly included or excluded reagent.

10.3 Filtering

In addition to selecting reagents with one of the selection strategies, you can filter the combinatorial results based on various predicted ADME properties of the molecules. These properties are generated by QikProp. Several filters are provided, and you can customize these filters by including or excluding any of the listed properties and by setting the limits on the acceptable range of the properties. The property ranges in the filters represent the undesirable property ranges: molecules whose properties fall inside these ranges will be filtered out. The filters are set up in the Filter tab of the Filter and Select dialog box.

If you apply both a filter and a selection strategy, the filtering is performed first, then the selection strategy is applied to the results of the filtering. Manual selection overrides both filtering and the selection strategy.

The Filter set option menu provides three preset filters, Druglike, Leadlike, and Coarse, a customizable filter (Custom), and an option to bypass filtering (None). If you modify one of the preset filters, the menu item switches to Custom.

Filtering is not available unless you have a QikProp license, and have combinatorial docking results.

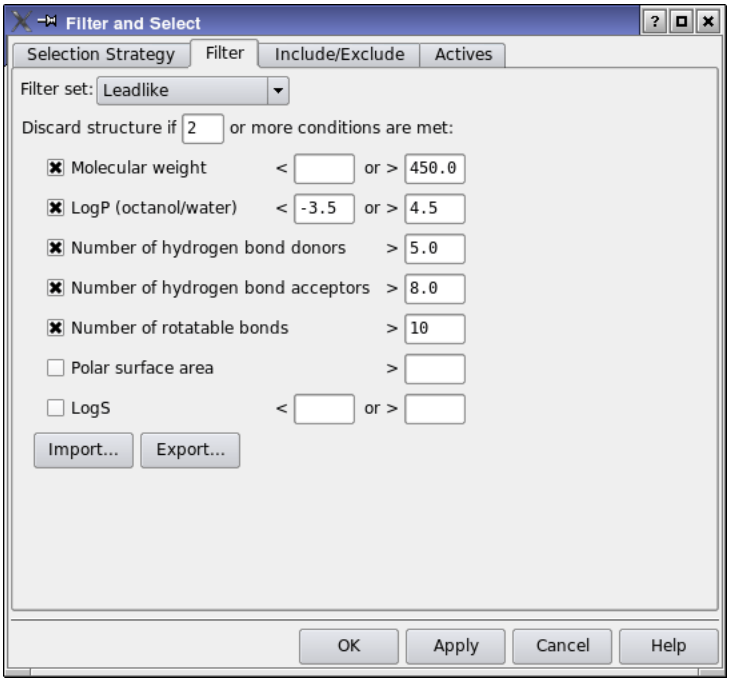


Figure 10.3. The Filter tab of the Filter and Select dialog box.

To use a predefined filter:

1. Choose a filter from the Filter set option menu.

To use a custom filter:

1. Choose a filter set from the Filter set option menu.

   Choose Custom for a blank filter form, or choose one of the predefined filter sets to alter.

2. Enter the number of conditions that must be met for the molecule to be discarded in the Discard structure if *N* or more conditions are met text box.
3. Select the properties that you want to include in the filter set.
4. Enter the values in the text box that define the property range.

10.4 Results of Selection and Filtering

The selection and filtering job is run when you click OK or Apply in the Filter and Select dialog box. After the job finishes, the results of the selection and filtering process are displayed in the table in the upper part of the panel. The results—the current filter and selection settings and the results from these settings—constitute a library selection. The library size is displayed next to the Create Library button. The first four columns of the table are identical to the corresponding columns of the Reagent retention table from the Dock Library step. The columns are described in Table 10.2.

Table 10.2. Description of the results table.

Column	Description
Name	Attachment position label
Reagent File	Name of the reagent file (minus the extension).
Single	Number of reagents from this file that survived the single-position docking.
Comb	Number of reagents from this file that survived the combinatorial docking. This number includes all reagents that contributed to any structure that docked well.
Library	Number of reagents selected for the library.
Best	Best GlideScore of the molecules used to select reagents at this position for the library. If combinatorial results are used in the selection, the best GlideScore is the same for all reagents.
Worst	Worst GlideScore of the molecules used to select reagents at this position for the library.

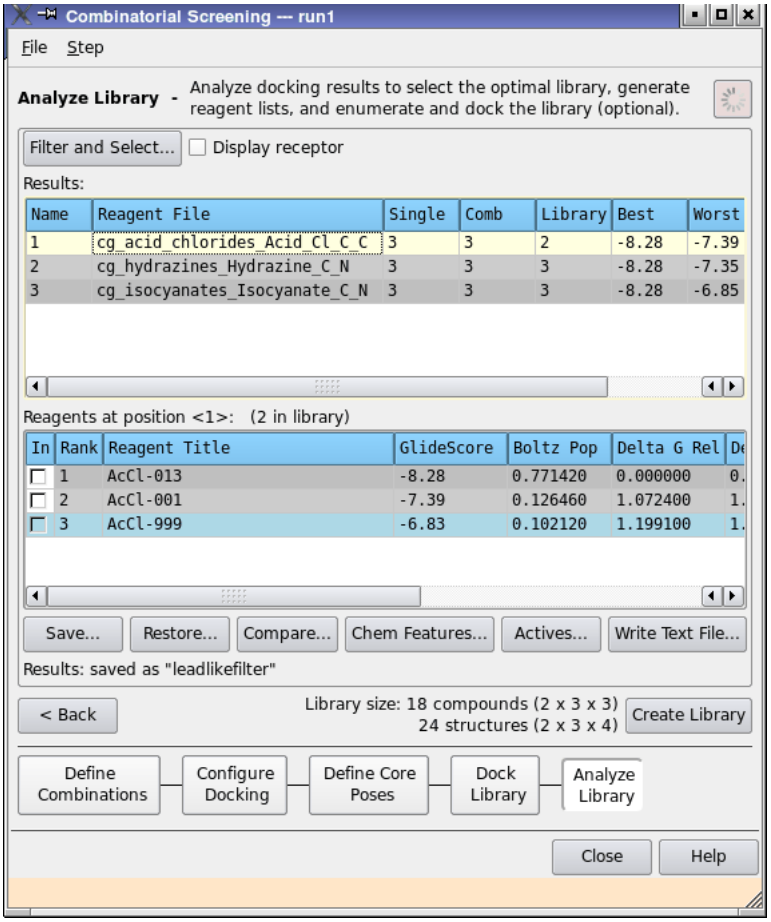


Figure 10.4. The Analyze Library step.

When you select a row in the results table, the reagent list with results for that position is displayed in the Reagents at position *N* table. This table displays the top reagents for the position selected in the results table along with various properties (see Table 10.3). The list includes both the reagents that were chosen for the library at this position and a number of the best reagents that were not chosen. The reagents that were not chosen for the library are highlighted in blue.

In addition to examining the numerical results of the selection and filtering, you can view the best-scoring structure for each reagent in the Workspace. To do so, select the desired position in the results table, then click the In column of the Reagents at position *N* table for the reagent that you are interested in. The best-scoring singly substituted or fully substituted structure that

Table 10.3. Description of the Reagents at position table.

Column	Description
In	Workspace inclusion status. You can use this column to display the best-scoring molecule for that reagent in the Workspace. However, manually included reagents that did not contribute to any docked molecule in the docking stage cannot be included in the Workspace.
Rank	Numerical rank of the reagent. The reagents are listed in order of rank; the rank is determined by the lowest GlideScore of any molecule that contains this reagent. A + sign indicates that the reagent was manually selected for inclusion in the focused library. Manually included reagents are listed at the top and are assigned the top ranks.
Reagent Title	Title of the reagent. This is the title chosen in the Reagent Preparation panel.
GlideScore	Best GlideScore for any compound in the library that used this reagent.
Boltz Pop	Boltzmann population for the reagent. Boltzmann population strategy only.
Delta G Rel	Effective relative GlideScore for this reagent treated as a free energy, derived from the Boltzmann populations. The best reagent has a Delta G Rel of zero. Boltzmann population strategy only.
Delta G Abs	Effective absolute GlideScore for this reagent treated as a free energy, derived from the Boltzmann populations. Boltzmann population strategy only.

contains the reagent is displayed. If you want to view the structure with the receptor, click Display Receptor. To remove the receptor, clear the Workspace.

If you decide that you want to include or exclude a reagent at a particular position, right-click in the row of the Reagents at position *N* table for the reagent and select Add to Include List or Add to Exclude List from the shortcut menu that is displayed. Selecting a menu item opens the Filter and Select dialog box so that you can re-run the library selection job with the newly included or excluded reagent.

Running a new selection and filtering job overwrites the results of the current job (the current library selection). You can save the results in the run, by clicking Save. The Save Results dialog box is displayed, in which you can name the library selection. Once you have saved a library selection, you can restore it later as the current selection by clicking Restore, then selecting the library selection in the Restore Results dialog box and clicking OK. You can also delete a library selection in the Restore Results dialog box by selecting it and clicking Delete. The name of the current library selection is given immediately above the Save button—or if it is not saved, a message to indicate that has not been saved.

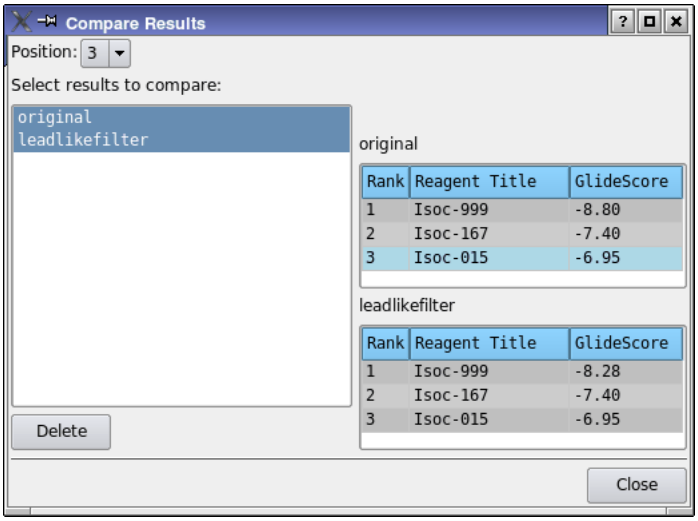


Figure 10.5. The Compare Results dialog box.

Saved library selections are stored within the run, and are not available outside the project. To save a library selection to an external file, click Write Text File, and navigate to the desired location in the file selector that is displayed. The contents of the file include the options used to run the library selection job, and the Reagents at position *N* table for each attachment position.

10.5 Comparing Results

In the process of selecting the optimal library, you may want to compare several library selections. To do so, these definitions must be saved in the run, using the Save button. To compare the saved library selections, click Compare, then in the Compare Results dialog box, select the library selections in the Select results to compare table. You can select and deselect items in this table with the usual click, shift-click and control-click actions. As you select the item, the Reagents at position *N* table for that library selection and the chosen position is displayed to the right of the list. To change the position for which results are displayed, choose the position from the Position menu. All the tables are updated with the results for the new position. You can compare up to six library selections.

You can also delete library selections, by selecting them in the Select results to compare list and clicking Delete. The selections are deleted without further confirmation, so check your choices carefully.

10.6 Creating a Library

When you have decided on the library selection, you can proceed to create the library. The library is a Maestro file that contains structures generated from the reagent variants, subject to a GlideScore threshold. A reagent variant is discarded if the GlideScore of the best-scoring molecule to which it contributed is above the threshold. The threshold is the highest GlideScore of any molecule that contributed a reagent to the library. In addition, a plain text file is written that contains a summary of the library selection. This is the same file as is written with the Write Text File button. The library can be created with or without docking the results.

If the library is docked, the output Maestro file includes the receptor and the docked structures. The library can be docked with any of the Glide docking modes, but you must have a separate Glide license. If you choose Glide SP docking or Glide HTVS docking, the Post-docking section becomes available, and you can select options for post-docking minimization and applying strain correction terms. For more information on these options, see Section 5.7.2 of the *Glide User Manual*.

If the library is enumerated only, you can choose to untangle and minimize the structures. When the structures are built, it is possible that the structures are "tangled"—for example, a chain might go through the center of a ring. In CombiGlide you can choose to untangle these structures, and then minimize them. Untangling and minimizing the structures takes longer, but the results are superior. Untangling is performed by default in the Dock Library step when the fully substituted structures are created.

To create the library:

1. Click Create Library.

   The Create Library - Options dialog box opens.

2. Choose a mode for creating the library.

   You can select Enumerate only, do not dock, or Enumerate and dock with.

3. Select options for the library creation mode.

4. Click OK.

   A Start dialog box opens.

5. Select job options, and click Start.

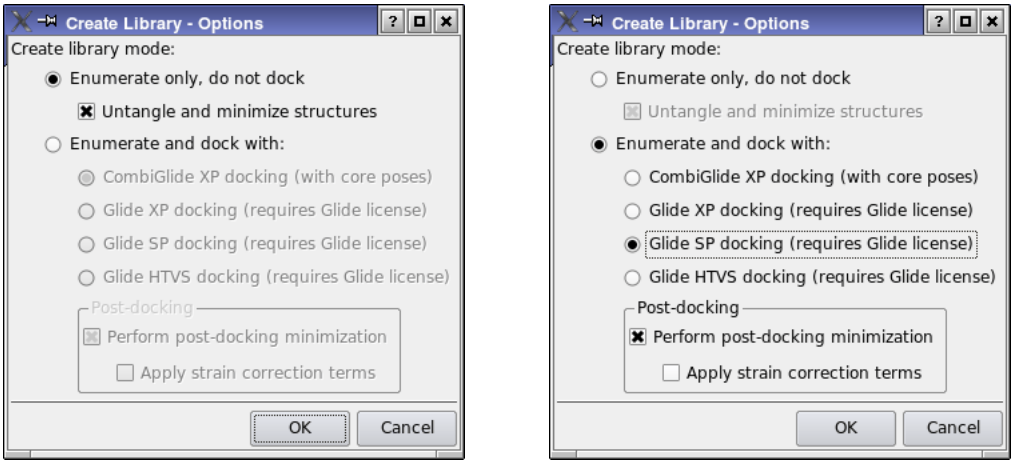


Figure 10.6. The Create Library - Options dialog box showing options for the two modes.

Combinatorial Screening: Analyzing the Library

When you have selected a small library of compounds that is likely to bind well, you may want more information about the library than is given by the GlideScore or the filtering. For example, the full library might include known actives, and you want to know how well the focused library recovers these actives. Or you might want to know what kinds of chemical features at each attachment position are found in the compounds that bind well. CombiGlide provides several tools for analyzing the results of a combinatorial docking run to obtain insight into what contributes to binding.

11.1 Chemical Feature Analysis

When selecting a small, high-ranking combinatorial library, you may be more interested in trends than in specific reagents. For example, you might like to know if hydrogen-bond donors are more likely than average to occur at a particular attachment point. These questions are addressed by the chemical feature analysis capabilities of CombiGlide.

Chemical feature analysis is done by classifying functional groups in side chains into chemical feature categories. The classification of the functional groups is performed automatically in the reagent preparation step, and the information is added to the reagent files. This classification is used later when a selection is performed using the Filter and Select dialog box. As part of the selection process, the chemical feature analysis is performed. The library must be one that was generated from a full combinatorial docking job: single-position results cannot be used.

When you have run a selection job, you can click Chem Features in the Analyze Library step to open the Chemical Features panel, which presents the results in graphical form. The panel and its display modes are described in subsections below.

If you want to apply chemical feature analysis to a CombiGlide 1.0 run, you can perform the classification on the reagent files with the command-line tool `cg_add_chem_features`, which is described in Appendix B. Once you have converted the reagent files, you can go to the Analyze Library step to run the chemical feature analysis, without having to repeat the entire docking run. When you rerun the selection process from the Filter and Select dialog box, chemical feature results are added to the output.

The results are more likely to be meaningful when a small library is created in the Filter and Select run—say, a 1000-compound library, if the virtual space has size 1M. This is because the strongest binders, which will predominate in a small library, can be expected to most strongly

discriminate in favor of specific chemical features. On the other hand, a library that is too small will not provide a good statistical sampling of good binders.

11.1.1 Features and Types

Chemical features can be classified into one of the following categories:

- A—hydrogen bond acceptor
- D—hydrogen bond donor
- H—hydrophobe
- N—negatively charged center
- P—positively charged center
- R—aromatic

These categories reflect standard pharmacophore types, as used by Phase, for example. However, some of the chemical feature definitions used by CombiGlide differ from standard pharmacophore definitions.

- Classification as N and P is only done when the structure is actually in the ionic form. This is because CombiGlide docks the charged and uncharged structures separately.
- A side chain (the part of a reagent that remains after removing the reactive functional group that is discarded when it is bound to the core) may bear multiple features.
- A side chain is considered to be hydrophobic only if it is hydrophobic overall, not if it bears only a few connected nonpolar carbons. This definition is designed to weed out small hydrophobes in otherwise hydrophilic molecules. A structure is considered hydrophobic if any of the following conditions is met:
    a. It has only hydrophobic features.
    b. It has hydrophobic features containing five or more atoms. This includes most aromatic hydrophobes.
    c. It has a QikProp-calculated logPo/w greater than some cutoff value (2.5 by default) and also contains a hydrophobe, however small. This condition prevents polar molecules containing a few hydrophobic carbon atoms from being considered hydropobic. The QikProp calculation is performed by replacing the reactive functional group in the side chain with a methyl group.
- A single group in a side chain can belong to multiple feature categories. For example, carboxylates are acceptors (A) as well as negative centers (N), and a hydrophobic aromatic group is both hydrophobic (H) and aromatic (A).

We define a *type* as the list of features that a side chain has. For example, a side chain that has A, D and H features (one or more of each) will have type ADH. Because every side chain has

exactly one type, the sum of the fractions of side chains bearing each type is unity. This permits several analyses that are not available for features.

Feature and type analysis are both based on comparison between frequencies in the reagent collection used to define the full virtual library at a given position on the core (the initial collection) and the frequencies that appear at the same position in the selected library (the final collection).

The broader the distribution of types in the full library, the more discrimination it is theoretically possible to observe in the final collection. For example, if the side chains in the initial collection contain only hydrophobes, the selected library can contain only hydrophobes, and no enrichment would be visible. Thus, chemical feature analysis is not useful if the initial reagent collection is highly focused around chemical features and types known to be active at the site in question.

11.1.2 Displaying Feature and Type Analyses

Three analyses are presented in the Chemical Features panel, which can be selected with the options at the top of the panel:

- Side Chain Features
- Overall Type Selectivity
- Type Preferences

These analyses are described in the following subsections.

You can save the figures that are displayed in the panel in PostScript or GIF format, by making the relevant choice from the File menu. Each file contains only one figure—the one that is displayed—so you must display each figure in turn and save it to get a complete set of figures for all positions.

You can also export the data on which the figures are based either as a CSV file or a plain text file, again by making the relevant choice from the File menu.

11.1.2.1 Side Chain Features

The initial and final feature frequencies are displayed side by side. Error bars show the standard error, based on counting statistics. When the two frequencies differ by more than one standard error, the difference is significant. The results shown below indicate strong enrichment of acceptors, donors, and positive features for the position shown.

From this chart you can also obtain information on elimination of features, or on requirements for features. If the fractional appearance for the selected library is zero, the feature has been eliminated; if it is one, the feature is required.

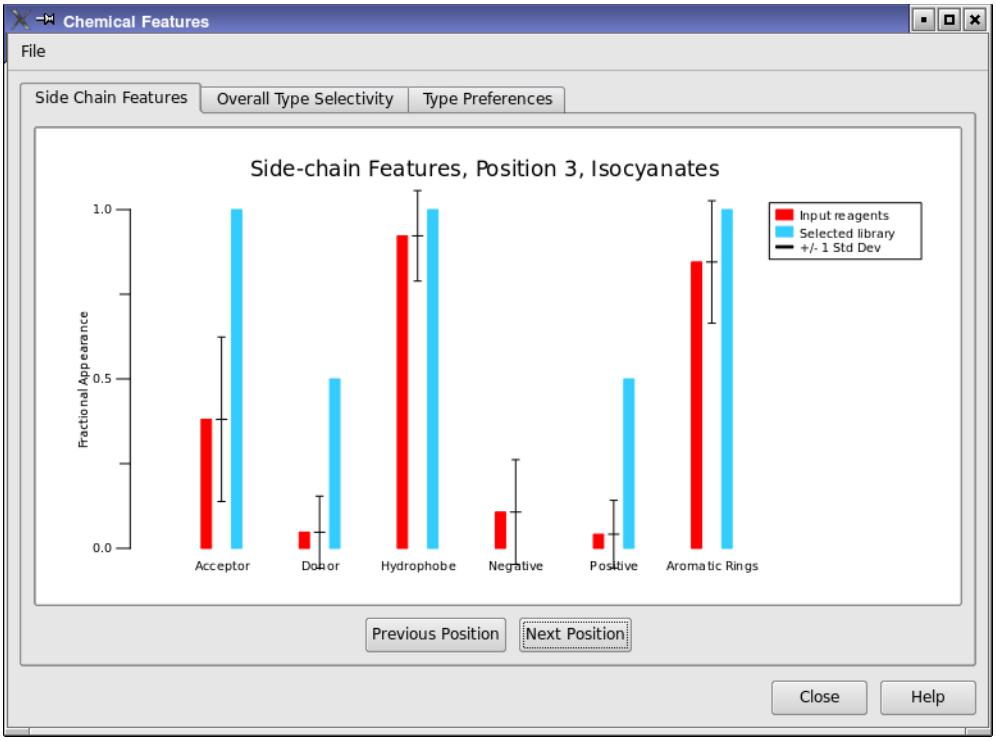


Figure 11.1. Chemical Features panel, showing Side Chain Features chart.

To change the position, click Next Position or Previous Position. These buttons cycle through the attachment positions and display the chart for that position.

11.1.2.2 Type Selectivity

Entropy-based statistics can be used for fractional compositions that sum to unity, such as for the type data. We use a measure based on the Kullback-Liebler divergence to compute the selectivity of each position on the core. If a position has a selectivity of N, it is similar to saying that only one out of N types is selected at that position. This value would in fact apply if the types were uniformly distributed in both the initial and the final distribution, but the initial side-chain collection contained N times as many types as the final one.

In the following example, we can see that position 3 of this system has a selectivity of 7, which is much greater than the selectivity observed at the other positions, which is close to 1. Position 3 is the position for which the feature data were shown earlier. It is the high degree of discrimination in types that leads to the high selectivity.

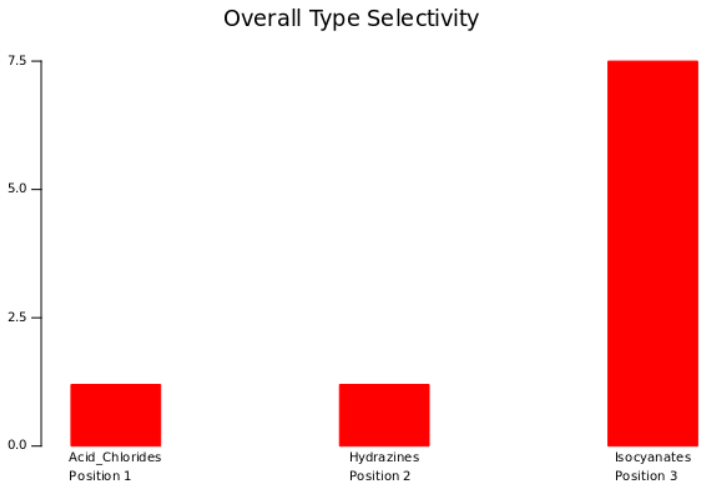


Figure 11.2. Overall type selectivity chart.

Selectivity is always relative to the diversity of the reagent set used. If the input reagent set is already narrowly focused on the reagent types that bind, the observed overall selectivity will be low, even if the position is highly selective relative to all possible chemical types.

For positions with low selectivity, the type selectivity details can help determine whether the low selectivity is due to low diversity in the reagent set or to greater discrimination at other positions. A simple measure is obtained from the number of types that appear in a reagent list. More types means more diversity. The Type Selectivity Details chart, which you can display by clicking Show Type Details, shows the number of types in the input reagent collections in pale blue and the number of types in the selected library in bright blue. An example is given in Figure 11.3.

In the example used for the figures, it can be seen that by this measure, the input reagent collections used at positions 1 and 2 are about as diverse than those used at position 3. Thus, using this measure of diversity, it could be concluded that the greater overall type selectivity exhibited by position 3 is intrinsic, rather than due to the use of a more diverse reagent collection at that position.

A better measure of diversity takes into account the non-uniform distribution of types at a position. For example, a two-type collection is more diverse if the types occur with equal frequency than if one type occurs 99% of the time and the other occurs 1% of the time. The latter situation is little different from having only a single type present. Such a measure that is based on the statistical entropy is also included in the type selectivity details. The entropy is converted into an effective number of types. These effective numbers are displayed in light green (for the input set) and dark green (for the selected library).

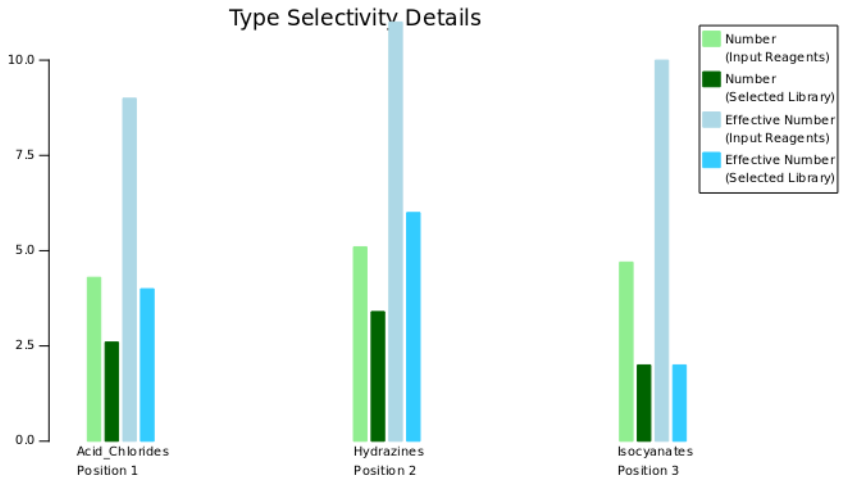


Figure 11.3. Type selectivity details chart.

For the example above, the effective number of types does not change much between the positions, so again it can be concluded that the high overall type selectivity exhibited at Position 3 is intrinsic, rather than artifactual.

11.1.2.3 Type Preferences

This chart visualizes how the distribution of chemical types changes between the initial and final collections. The upper bar in the display represents the initial side-chain collection and the lower bar represents the final side-chain collection. The width is normalized to 100% in both cases. Each vertical segment represents a chemical type, and each type has horizontal bars colored in for each feature that is part of the type; the width of each type represents its fractional appearance. For example, in the chart shown, the type farthest to the left, which we designate ADHPR, contains acceptors, donors, hydrophobes, positive centers and aromatics, but no negative feature. This type comprises only a few percent of the initial side-chain collection but about 50% of the side chains in the selected library. The third type, which may be designated AHR, is also strongly enriched in the selected library.

You can change the sort order by selecting one of the three options in the lower part of the panel: Alpha Sort sorts the types in alphabetical order of the type label; Input Reagents Fraction Sort and Selected Library Fraction Sort sort in order of fractional appearance in the respective collection.

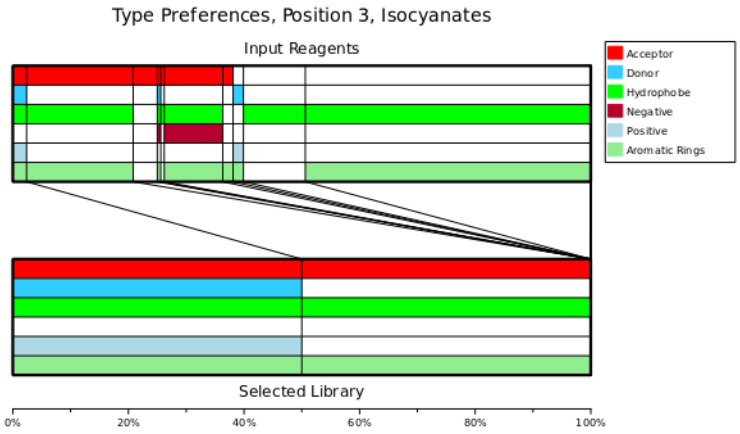


Figure 11.4. Type preferences plot.

11.1.3 Using Chemical Feature Analysis Results

The results of a chemical feature analysis can be used in library refinement. Based on results with a diverse side-chain collection, a second-generation library can be designed that is enriched in the types and features shown to dominate at the various positions. Positions that exhibit low selectivity are less sensitive to the types and features that appear there, and a possible strategy is thus to use larger reagent collections (in proportion to selectivity values) at the more selective positions.

Two warnings, one pertaining to focused collections and one pertaining to diverse collections, must accompany this view:

- If the initial side-chain collection is already highly focused, it will not be possible to observe strong enrichment or high selectivity.
- If a diverse side-chain collection is too small, it might not be possible to find enough good binders at all positions to obtain meaningful results, due to inadequate sampling of chemical space. This will also lead to low enrichment and selectivity.

11.2 Active Compound Analysis

If you are interested in how well the focused library picks up known actives, you can use the active compound analysis capabilities to generate enrichment data and an enrichment plot. The analysis is done when you run a selection job from the Filter and Select dialog box.

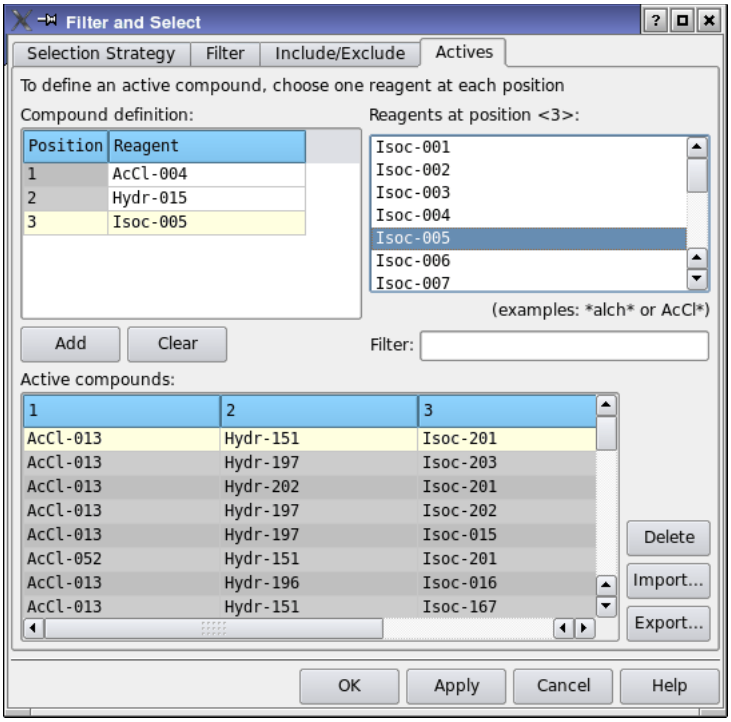


Figure 11.5. The Actives tab of the Filter and Select dialog box.

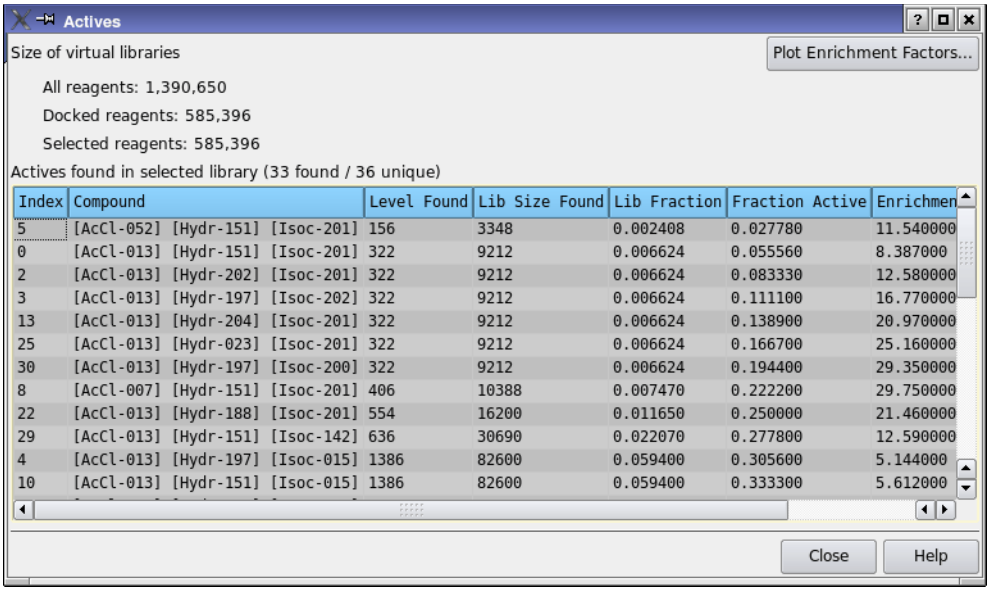
The actives are specified in terms of a selection of the reagents at each position—they must, naturally, be members of the fully enumerated library in order to calculate an enrichment. The selection is made in the Actives tab of the Filter and Select dialog box, as follows:

1. Choose a position in the Compound definition table.
2. Choose a reagent from the Reagents at position *N* list.

   You can filter the reagents by entering text in the Filter text box, with * to represent an arbitrary string and ? to represent an arbitrary single character. You can also specify alternatives by listing them in braces { }, separated by commas. For example, 3-{F,Cl,Br}* would match any reagent that began with 3-F, 3-Cl, or 3-Br. All reagents whose titles contain the specified string are listed in the Reagents at position *N* list when you press ENTER.

   You can also edit the Reagent column in the Compound definition table directly, but the name you enter must be one on the list. Entering text sets the filter in the Filter text box.

   If you want to clear the Reagent column at the selected position, click Clear.



Index	Compound	Level Found	Lib Size Found	Lib Fraction	Fraction Active	Enrichmen
5	[AcCl-052] [Hydr-151] [Isoc-201]	156	3348	0.002408	0.027780	11.540000
0	[AcCl-013] [Hydr-151] [Isoc-201]	322	9212	0.006624	0.055560	8.387000
2	[AcCl-013] [Hydr-202] [Isoc-201]	322	9212	0.006624	0.083330	12.580000
3	[AcCl-013] [Hydr-197] [Isoc-202]	322	9212	0.006624	0.111100	16.770000
13	[AcCl-013] [Hydr-204] [Isoc-201]	322	9212	0.006624	0.138900	20.970000
25	[AcCl-013] [Hydr-023] [Isoc-201]	322	9212	0.006624	0.166700	25.160000
30	[AcCl-013] [Hydr-197] [Isoc-200]	322	9212	0.006624	0.194400	29.350000
8	[AcCl-007] [Hydr-151] [Isoc-201]	406	10388	0.007470	0.222200	29.750000
22	[AcCl-013] [Hydr-188] [Isoc-201]	554	16200	0.011650	0.250000	21.460000
29	[AcCl-013] [Hydr-151] [Isoc-142]	636	30690	0.022070	0.277800	12.590000
4	[AcCl-013] [Hydr-197] [Isoc-015]	1386	82600	0.059400	0.305600	5.144000
10	[AcCl-013] [Hydr-151] [Isoc-015]	1386	82600	0.059400	0.333300	5.612000

Figure 11.6. Actives dialog box.

3. Repeat Step 1 and Step 2 for the remaining positions.
4. Click Add.

   A row is added to the Active compounds table. There is one column in this table for each position, which lists the reagent used at this position to define the active compound.

When you have finished defining the actives, you can run the library selection. Following the run, you can view information on the recovery of actives by clicking the Actives button in the Analyze Library step. This button opens the Actives dialog box, which shows the results in tabular form. The columns of the table are described in Table 11.1. From this dialog box, clicking Plot Enrichment Factors displays a standard enrichment plot, in the Active Plot panel. The panel presents of a plot of the fraction of actives found against the fraction of the database screened, along with a plot of enrichment factors against the size of the library.

This plot only extends to the range of the library selected. To see a full enrichment curve, set Max library size in the Filter and Select dialog box to a size exceeding that of the entire virtual library, then display the enrichment plot based on these results. This plot will not extend to the full virtual space, but it will cover the range accessible from the reagents used in the combinatorial docking (allnode) run.

An example of an enrichment plot is shown in Figure 11.7.

Table 11.1. Description of columns in the Actives found in selected library table.

Column	Description
Index	Index number of the active compound in the list of actives.
Compound	Identifier of the active compound, given in terms of the reagents used.
Level Found	Level in the selection procedure where this active compound was included in the library.
Lib Size Found	Size of the selected library when this active compound was found.
Lib Fraction	Fraction of the virtual library that was screened when this active compound was found. Equal to Lib Size Found divided by the total virtual library size (All reagents value in the Size of virtual libraries text area.
Fraction Active	Fraction of actives identified (including this compound) when this compound was identified in the selection procedure.
Enrichment	Enrichment factor, equal to the ratio of the fraction of actives identified (Fraction Active) to the library size ratio (Lib Fraction).

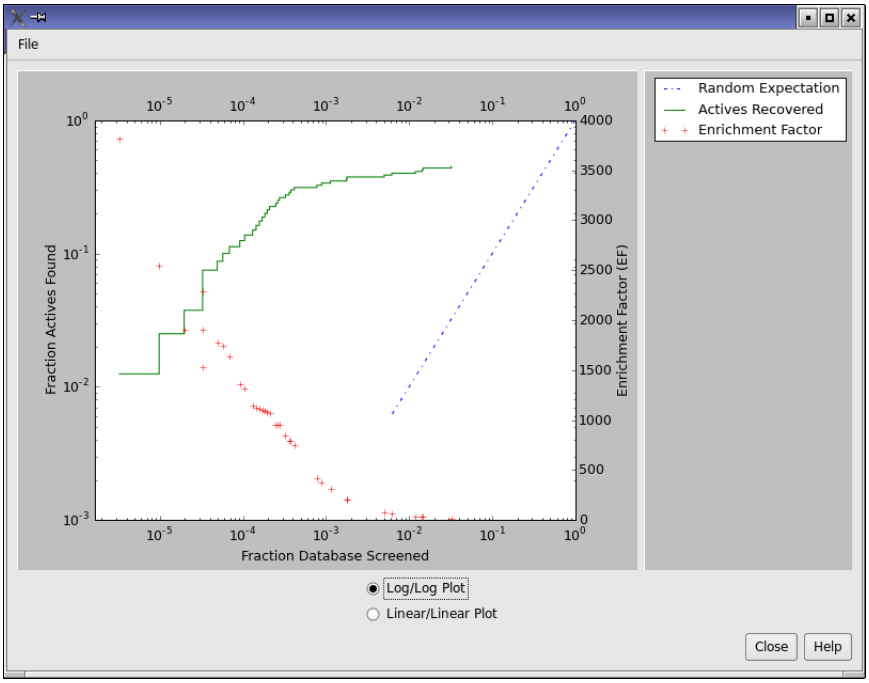


Figure 11.7. Enrichment factor plot.

Running Tasks from the Command Line

This chapter describes how to run various CombiGlide tasks from the command line.

12.1 Reagent Preparation

Reagent preparation is implemented as a set of additions to LigPrep, which includes two scripts, `vcsprep` and `reagentprep`, and a command-line option to the `ligprep` command. The script `vcsprep` is run at the beginning of the LigPrep workflow, to ensure that the titles of the compounds are unique, and again at the end, to generate the statistics file and the SQLite database file that are used by CombiGlide. The script `reagentprep` is run on the structures produced by LigPrep, to add the labeling of the attachment bond. The concept of reagent preparation is described in detail in Section 5.2 on page 38.

To prepare a single reagent file for use in the combinatorial screening workflow or in combinatorial library enumeration, you can use the `ligprep` command as follows:

`$SCHRODINGER/ligprep` *ligprep-options* `-vcs` *functional-group* `-o` *jobname*`.bld`

The *ligprep-options* are the usual LigPrep structure preparation options, which are described in Table 2.1 of the *LigPrep User Manual*. The `-vcs` option specifies the name of the functional group to use. The names of the standard functional groups supplied with CombiGlide are the long names given in Table 5.1 on page 41. You can also obtain a list of names with the following command:

```
$SCHRODINGER/utilities/reagentprep list
```

Information on providing custom functional groups is given in Appendix A. The output file must have a `.bld` extension, which indicates among other things that the file contains the information on the attachment bond (“grow bond”). The input file can be any file accepted by LigPrep.

You can also distribute the reagent preparation job by using the `para_ligprep` command instead of `ligprep`. However, because there is no checking for duplicates between subjobs run by `para_ligprep`, the procedure used to ensure that titles are unique is to append the subjob number to each title, in addition to adding a suffix to duplicate titles. If this behavior is not acceptable, you should use `ligprep` instead.

In addition to the main output file, three structure files are produced by the `reagentprep` stage that contain structures that were rejected:

- *jobname*_rp-mult.mae contains structures that have multiple instances of the functional group specified that are not equivalent. Structures that have equivalent instances, such as the carboxyl groups in adipic acid for a carboxylic acid functional group, are accepted and one instance is labeled. Inequivalent instances, such as the carboxyl groups in glutamic acid, are rejected.
- *jobname*_rp-nomatch.mae contains structures that did not match the functional-group specification. If the input file should contain structures that all match, this file should be checked to determine why the structures did not match. If the input file is a master file of reagents for all functional groups, and reagent preparation is used to filter the structures, then this file merely contains the structures that did not match.
- *jobname*_rp-ring.mae contains structures for which the bond to be broken occurs in a ring. The Schrodinger build system cannot at this point handle such structures.

You should not use the output from a reagent preparation job as input to another job. If you do, the job will fail. If you want to label the reagents differently, you should start with the input file from the original job.

12.2 Combinatorial Library Enumeration

Enumeration of a combinatorial library can be done with the program combgen. The library can be constructed from a prepared core-containing structure, in which substitutions are made at the defined attachment points, or it can be constructed by linking several fragments in a linear sequence. The substitutions on a core-containing structure can also consist of several fragments.

The fragment libraries used to generate the combinatorial library can be taken from the standard Maestro builder (the default), or from reagent libraries prepared with reagentprep. If you want to use both, you must copy the libraries to a common location. Both the .bld and the .frg files must be copied, and the .ini file must be modified to include all desired libraries.

12.2.1 The combgen Command

The syntax of the combgen command is as follows:

$SCHRODINGER/combgen *jobname* [*options*]

The options are described in Table 12.1.

Table 12.1. Options for the combgen command.

Option	Description
`-h[elp]`	Show brief usage message.
`-doc`	Show longer usage message.
`-l[og[file]]` *logfile*	Specify log file, which reports the progress of the library generation. Use `-l -` to send the log output to the terminal window. Default: *jobname*`.log`.
`-i[n[file]]` *infile*	Command file defining the combinatorial library. Default: *jobname*`.in`. The syntax of this file is described below.
`-o[ut[file]]` *outfile*	File of generated library member molecules, in Maestro format. Default: *jobname*`.mae`.
`-r[eport]` *nsec*	Report on progress of library generation every *nsec* seconds.
`-s[um[mary]]`	Report combinatorial library summary information and exit. No molecules are generated.
`-n|-index|-range` *indexrange*	Build members across the supplied index range, which must be in the format *n1*`:`*n2*.
`-c[omp_vec]` *compvec*	Build a single library member by composition vector. The composition vector is a comma-separated list of fragment indices, one per attachment point, e.g. `-c 13,4,2,22`.
`-C|-comp_vec_list` *listfile*	Build a set of members according to the composition vector list in the file specified. The list must contain one composition vector per line. Each vector is a comma-separated list of fragment indices.
`-m[mfrag_init]` *initfile*	Specify the file that lists the fragment library names. The file must contain one library name per line, which is the name of the `.bld` file minus the extension. The default file is `mmfrag.ini`, which is stored in `$SCHRODINGER/mmshare-v`*version*`/data/res`. This library contains all the fragments used in the Build panel. In order to use your own prepared reagents, you must list the names for your reagent files in a file and specify that file with this option. The fragment libraries and the file that lists the libraries must be in the same directory.
`-mq|-maxq|-maxabstotq` *fcharge*	Maximum absolute value of the (total) formal charge on any library member.
`-mi|-maxions|-maxtotabsq` *numions*	Maximum value for any library member of the sum of the absolute value of the formal charges on each atom.
`-minmw|-minmolwt` *minmw*	Minimum molecular weight of any library member.
`-maxmw|-maxmolwt` *maxmw*	Maximum molecular weight of any library member.

Table 12.1. Options for the combgen command. (Continued)

Option	Description
`-mina[toms]` *minat*	Minimum number of atoms in any library member.
`-maxa[toms]` *maxat*	Maximum number of atoms in any library member.
`-minh|-minheavy` *minh*	Minimum number of heavy (non-hydrogen) atoms in any library member.
`-maxh|-maxheavy` *maxh*	Maximum number of heavy (non-hydrogen) atoms in any library member.
`-unt[angle]`	Untangle generated structures. Runs `ligprep -unt`.
`-runt|-retunt`	Return structures from before and after the untangling stage.
`-NJOBS` *subjobs*	Number of subjobs for the untangling stage.
`-NO_REDIRECT`	Do not run under Job Control.

12.2.2 The combgen Input File

The specification of the combinatorial library is made in the input file, which consists of a set of command lines. Each valid command line consists of a command name, followed by a colon (:), followed by a list of parameters specific to the command. The command names are case-insensitive. Comments can be added to the file after a # character.

There are four commands: `Core`, `Chain`, `Frag`, and `Vlib`.

The first command must be a `Core` command, and there must be only one `Core` command in the input file. The `Core` command can be followed by one or more `Chain` commands. Each `Chain` command should be followed by the appropriate number of `Frag` commands. `Chain` and `Frag` commands can be interspersed as long as each `Frag` command is preceded by the `Chain` command it refers to.

`Vlib` commands are optional. They can be used to specify and report on virtual fragment libraries, which are defined as subsets of the actual, full fragment libraries. Any `Vlib` commands that initiate or alter virtual fragment library fragment subsets must precede the first `Frag` command in the input. These `Vlib` commands are typically be placed at the top of the file. Any `Vlib` commands that merely report on fragment subsets can occur after `Frag` commands, and are typically placed at the end of the input file.

The structure of a typical input file can be symbolically represented as follows:

```
Core
Vlib
...
```

```
Vlib
Chain
Frag
...
Frag
Chain
Frag
...
Frag
...
Vlib
...
Vlib
```

Annotated examples of input files are provided in `$SCHRODINGER/combiglide-v`*version*`/samples`.

The commands with their syntax and use are described in the following subsections.

12.2.2.1 The Core Command

The `Core` command specifies the core-containing molecule, if one is to be used. The syntax is as follows:

`Core:` *filename*

The file must be in Maestro format, and the first structure from this file is taken as the core-containing molecule. If no file name is given, a coreless chain is assumed.

The structure in the file must include the grow name labels required to specify attachment points. These names specify the "from" atom (the atom from which the fragment is grown), and the "to atom" (the atom towards which the fragment is grown, and which is replaced, along with any atoms attached to it, when the fragment is added). The convention to use for these atoms is *n*`f` for the "from" atom and *n*`t` for the "to" atom, where *n* is the index of the attachment point, starting from 1. This is the convention that is applied when you label a core-containing molecule in Maestro. If you want to add these names manually, you should add them to the `s_m_grow_name` field in the atom block of the Maestro file.

12.2.2.2 The Chain Command

The `Chain` command specifies the chain to be attached at a given attachment point on the core-containing molecule. The chain can consist of multiple fragments, joined in a linear sequence.

`Chain:` *chain-name min-frags max-frags from-name to-name*

The fourth and fifth parameters apply only when a core has been specified. The parameters are described below.

chain-name	Name of the chain. This name is used by `Frag` commands to refer to the chain. The name may not contain any spaces, even if the name is quoted.
min-frags	Minimum number of fragments to build in this chain in exhaustive combinations. The lowest allowed value is 0 when there is a core, or 1 for a coreless chain.
max-frags	Maximum number of fragments to build in this chain in exhaustive combinations. The lowest allowed value is *min-frags* (or 1 if *min-frags* is 0). If *min-frags* = *max-frags*, all combinations will have the same number of fragments substituted.
from-name	Grow name of the "from" atom of the core molecule bond to break when attaching the first fragment of the chain.
to-name	Grow name of the "to" atom of the core molecule bond to break when attaching the first fragment of the chain.

When there is no core, there must be exactly one `Chain` command. In this case, the first fragment functions as the core, with a single attachment point defined by the forward grow direction. Even though there is only one chain, it must be given a name, to which subsequent `Frag` commands can refer.

The following example specifies a chain named `ChainA`, with 1 to 2 fragments added, starting at the attachment point on the core defined by grow names `1f` and `1t`:

```
Chain: ChainA 1 2 1f 1t
```

The following coreless chain specification leads to a combinatorial library in which every member consists of 4 fragments bonded together:

```
Chain: dummy 4 4
```

12.2.2.3 The Frag Command

The `Frag` command specifies the fragment library that is to be used to attach fragments to the specified chain at the specified fragment number. There must be a `Frag` command for each fragment, from fragment 1 to *max-frags*, of each specified chain. The syntax of the `Frag` command is as follows:

`Frag` *chain-name frag-number lib-name*

The parameters of the `Frag` command are specified below:

chain-name Name of the chain to which this fragment will be attached. The name must match one of the names assigned in an earlier `Chain` command; the matching is case-insensitive.

frag-number Fragment number (position) in the chain. The number can range from 1 to the value of *max-frags* for the specified chain. Fragment numbers increase moving away from the core.

lib-name Name of the fragment library. All of the fragments in the named library are used in generating combinations. You can specify either an actual fragment library or a virtual fragment library (defined by a prior `Vlib` command). The library names of actual fragment libraries are those listed in the file specified with the `-m` command option (which by default is the file `$SCHRODINGER/mmshare-v`*version*`/data/res/mmfrag.ini`).

Fragment libraries that are used in a non-terminal location in the chain must have two grow bonds defined, a "from" bond and a "to" bond. The builder fragment libraries supplied with Maestro (in `$SCHRODINGER/mmshare-v`*version*`/data/res`) have two grow bonds defined. The reagent libraries prepared with `reagentprep` have only one grow bond defined, and must be used in a terminal location in the chain.

12.2.2.4 The Vlib Command

`Vlib` commands are used to specify virtual fragment libraries, which are defined to be subsets of normal, full fragment libraries. A virtual fragment library can be specified in a `Frag` command for the purpose of limiting fragment selections in combinatorial generations.

The syntax of the Vlib command is as follows:

`Vlib:` *subcommand parameters*

There are five subcommands, whose syntax and use is described below.

`Vlib: new` *vlib-name fraglib initialization*

The `new` subcommand creates a new virtual fragment library specification, and initializes the virtual library from an actual library. Subsequent `add` or `delete` subcommands define the subset of this library that is used in the virtual library.

vlib-name The name of the virtual fragment library. This name must not match an actual fragment library or another virtual fragment library.

fraglib The name of the actual fragment library of which this virtual fragment library is a subset.

initialization Keyword that specifies how to initialize the virtual fragment library. Valid values are:

`all` | `full`—Initialize the virtual library with all members of the full library.

`none` | `empty`—The virtual library is initially empty.

`Vlib: add` *vlib-name frag-name-list*

The `add` subcommand adds a specified list of fragments to the given virtual fragment library. The list is a blank-separated list of fragment names. Names of fragments that have already been added are ignored. You can include as many `add` subcommands as you want. This subcommand is normally used for a virtual fragment library that is initially empty.

`Vlib: remove` *vlib-name frag-name-list*

The `remove` subcommand removes a specified list of fragments from the given virtual fragment library. The list is a blank-separated list of fragment names. Names of fragments that have already been removed are ignored. You can include as many `remove` subcommands as you want. This subcommand is normally used for a virtual fragment library that is initially full.

`Vlib: list` *vlib-name*

The `list` subcommand displays the list of fragment names defined in the specified virtual fragment library. This subcommand must follow the `new` subcommand that defines the virtual fragment library. To generate a list of fragment names in a library, you can use this command immediately after a `Vlib: new` command with initialization to `all`.

`Vlib: count` *vlib-name*

The `count` subcommand displays the number of fragment names defined in the specified virtual fragment library. This subcommand must follow the `new` subcommand that defines the virtual fragment library.

All uses of the `Vlib` subcommands `new`, `add`, and `remove` must occur before the first `Frag` command. Uses of `list` or `count` can occur after the `Frag` commands.

When virtual fragment libraries are created, some temporary files are produced. These files are stored by default in `$HOME/.schrodinger/vlib_tmp`. You can specify the location for these files by setting the environment variable `SCHRODINGER_VLIB_TMPDIR`.

12.3 Protocore Preparation

The preparation of a single protocore file from a raw core file for use in core hopping can be done with the utility `protocore_prep`. The concept of protocore preparation is described in detail in Section 2.2 on page 9. The syntax of this command is as follows:

`$SCHRODINGER/utilities/protocore_prep` [*options*] *input-file output-file*

Both the input file and the output file must be Maestro files. The options are described in Table 12.2.

Table 12.2. Options for the protocore_prep command.

Option	Description
`-v[ersion]`	Display the program version and exit.
`-h[elp]`	Display the usage message and exit.
`-n` *GrowLength*	Set the maximum number of methylene linkers. The default is 0. Specifying more than two linkers is not recommended.
`-a`	Use the bonds that are already marked as attachment bonds in the input. By default, all nonpolar hydrogens are considered as attachment bonds.
`-p`	Use polar hydrogens. By default only nonpolar hydrogens are considered as attachment bonds.
`-e`	Grow linkers dendritically: the second and subsequent methylene insertions replace two hydrogens on the previous methylene rather than just one.
`-d` *level*	Set verbosity level (0-2)
`-l` *logfile*	Specify location of log file. The default is stdout.

Protocore docking can be run from the command line by writing out the input file, then using the `glide`, `impact`, or `para_glide` command to run the job. See Chapter 7 of the *Glide User Manual* for more information on these commands.

12.4 Library Selection

The selection of a full combinatorial library starting with a docked selection from a much larger virtual combinatorial space can be performed with the utility `libselector`. A full description of library selection is given in Chapter 10.

The `libselector` utility produces a log file and also a file containing reagents comprising the selected library. Libselector's input consists of N+2 SQLite database files, where N is the

number of substitutable positions on the core in the library definition. For each position, there is an SQLite database that is produced by the reagent preparation procedure along with the fragment files *filename*`.bld`; these are named *filename*`.sqlite`. If the structural output of the combinatorial docking job is *jobname*`.mae`, then the CombiGlide procedure should also produce *jobname*`.sqlite` and *jobname*`-comb.sqlite`, both of which `libselector` requires.

12.4.1 The libselector Command

The syntax of the libselector command is as follows:

```
$SCHRODINGER/utilities/libselector -s strategy -S strategy_options
    -j jobname [options]
```

The options are described in Table 12.3.

Table 12.3. Options for the libselector command.

Option	Description
`-s best|boltz`	Selection strategy. Required. `best`—base strategy on lowest GlideScore of docked structures. `boltz`—base strategy on Boltzmann sum derived from GlideScores of docked structures.
`-S` *strategy-options*	Options for the selection strategy. The strategy-options string is a comma-separated string of options with their values. The possible options depend on the selection strategy, and are described in Table 12.4.
`-j` *jobname*	Job name.
`-q` *qp-treatment*	QikProp treatment to use. The only available treatment currently is `filter`, which filters out input structures that do not meet the filtering criteria.
`-Q` *qp-treat-options*	QikProp treatment options. For `-q filter`, the options are `druglike`, `leadlike`, `coarse`, and *custom*. The custom treatment requires a file named *custom*`-qfilter.txt` to exist, which contains the filter criteria. The file format is described below.
`-I` *include-file*	File containing a list of reagents to be included at specified positions.
`-E` *exclude-file*	File containing a list of reagents to be excluded from specified positions.
`-A|-a` *filename*	File containing a list of actives in the selected library.
`-l` *log-file*	Log file.
`-r` *reagent-file*	Reagent output file

Table 12.3. Options for the libselector command.

Option	Description
`-v`	Sets verbose output.
`-h`	Display short usage message.
`-H`	Display long usage message.

Table 12.4. Selection strategy options

Option	Description
Options for all strategies	
`-Max,`*maxsize*	Maximum library size. Required.
`-max,`*n1*`:`*n2*`:...`	Maximum number of reagents at some or all position. If there are *N* positions, the argument must have *N*-1 colons. A missing number means that there is no maximum specification at the position concerned. For example, `-max,2::` would be a valid specification for a three-position library: a maximum of two reagents may appear at the first position and no maximum is specified for the second and third positions.
`-min,`*n1*`:`*n2*`:...`	Minimum number of reagents at some or all positions. The argument string is composed as for `-max`.
`-num,`*n1*`:`*n2*`:...`	Exact number of reagents at some or all positions. The argument string is composed as for `-max`.
Options for `best` *strategy*	
`-preempt`	Only applies when `-max` is specified for one or more positions. Once a position with a specified maximum number of reagents has its quota filled, structures from the input file that contain reagents not already selected at this position are removed from further consideration as other positions are filled. This could be an advantage if the contributions to binding of the reagents at their various positions are far from independent. On the other hand, it digs deeper into the list of input structures, discarding some with good GlideScores, and may even exhaust the list before the user's requested maximum library size is reached. Finally, it guarantees that even when positional maxima are specified, all structures used to determine the selected library are included in the selected library. (This condition is always satisfied when positional maxima are not specified.)

12.4.2 The QikProp Filter File

The custom QikProp filter file consists of a set of specification lines with one of the following forms

property-name min max
property-name op value

where property-name is the name of the QikProp property as it appears in a Maestro file, *min* and *max* are the minimum and maximum values for the property value, *op* is one of `gt` (greater than) or `lt` (less than), and *value* is a threshold. If the property value for a structure meets the specification, or the structure does not have the property, the structure passes the filter. The number of filter failures needed to reject a structure is given by a specification line in the file using the special property `i_qp_#rejections`. Comment lines may be included by beginning them with a `#` character.

A sample filter file follows:

```
i_qp_#rejections  gt     1
r_qp_mol_MW      150   650
r_qp_QPlogPo/w    -3     6
r_qp_PSA           gt  174.9999
```

12.4.3 Include and Exclude Files

The include and exclude files have a common format. The file consists of one or more lines that define a position; each such line is followed by one or more lines defining a reagent. The position definition line has the following format:

`position_name:` *name*

where *name* is the name of the position. The reagent definition lines contain the name of the reagent. Comments can be added on lines beginning with a `#` character. Blank lines are ignored.

12.4.4 Actives File

Actives files are specified with either the `-a` option or the `-A` option. These files provide a list of specifications of which compounds in the library are known active.

For an active file specified by `-a` *filename*, the specification of an active consists of a list of comma-separated integers. Each integer is the index of the reagent at the given position (starting with 1). in the corresponding `.bld` files used to define the library. Since the `.bld` files could contain multiple modifications of any reagent, the integers are necessarily specifying specific modifications. However, all modifications of a reagent are considered equivalent for the purpose of searching for actives. That is, the specification is understood to refer to the underlying reagent, not the specific modification.

For an active file specified by `-A`, the specification of an active consists of a list of reagent names, one for each position. Each reagent name is given in square brackets, so the following constitutes a valid example:

```
[3,4,5-trimethoxy-beta-phenylethyl amine] [acetic acid] [LMDR 426]
```

Within a pair of square brackets delimiting a reagent name, leading and trailing whitespace is ignored, but embedded whitespace is significant. Matching square brackets within a reagent name are accepted; unmatched embedded square brackets must be escaped with a backslash, thus: `\[`. In the two lines given below, the first specifies reagent name `g` whereas the second specifies reagent name `[ g ]`, where the white space is significant:

```
[   g ]
[ [   g ] ]
```

12.4.5 The Reagent File

The main purpose of this file is to provide lists of reagents that the program selected for each combinatorial position. In addition, it provides information about the overall run and the reagents, and also gives a list of "runners up" at each position: reagents that might be selected if the maximum library size were increased.

The file has a header section, which contains information about the run as a whole, followed by sections containing information about each grow position. Comments are included on lines that begin with a `#` character. Many lines are in the form of name-value pairs, where a name is a string followed by a colon. The name is followed by optional whitespace then by the value.

In the following description, the exact names used in the file are given only when they are not sufficiently descriptive to be obvious when seen.

12.4.5.1 Header Section

This section always includes `invocation:` and `args:` lines, giving the program name and arguments as it was executed. It also includes values for various library sizes:

- `lib_size:` size of the library selected by `libselector`
- `ext_size:` the maximum library size buildable from the output of CombiGlide docking
- `bld_size:` the size of the original virtual space.

These sizes are in terms of reagents, not reagent modifications; for example, acetic acid counts only once, even if it occurs at some position in the selected combinatorial library in both ionized forms.

If QikProp filters were in use, the filter values are given on `qp_filter:` lines.

If -a or -A was used to specify a list of known actives, statistics for the known actives are given:

- num_actives_unique: number of unique active combinations given in the input
- num_actives_found: specifies the number found in the selected library

For each active compound, its position in the input list (starting with 0) is given, followed by its composition, followed by a vector of values pertaining to it and when it was found when building up the library. Values include the level found, the size of the library when it was found, what fraction of the original virtual library this constitutes, what fraction of the specified actives was found when this one was found, and the enrichment factor (the ratio of the fraction of actives found to fraction of the original virtual library comprising the libselector selection when the active was found).

The "level found" corresponds to the step the active was first found at as the library was being built. For the best strategy, this is the index of the Glidescore-sorted CombiGlide docked structure where the active was first found, omitting structures skipped for reasons of QikProp filtering, user-specified exclusion, or preemptive strategy. For the boltz strategy, it is the level at which the reagent was found when traversing the list of reagents in order of decreasing Boltzmann fraction.

12.4.5.2 Position Sections

Each a position section starts with a position_name: line, then a reagent_filename: line, which specifies the name of the .bld file used to supply reagents at this position. A list of user-excluded reagent names at this position, if any, comes next.

Following this, the file gives the best and worst GlideScores of structures contributing to the reagent list that follows, and the number_selected:, which is the number of modifications of reagents selected at this position (where a modification could be an ionization state, a tautomerization state, or a stereoisomeric state).

Then come a set of lines, one for each selected reagent modification. From the point of view of synthesis, only the reagents matter, but from the point of view of later enumerating the selected virtual library, it is important to list all high-scoring modifications.

For each reagent selected, a vector of values is given that depends, to some extent, on the strategy used. The level at which each reagent was found is given (see discussion above concerning the known actives in the header section); this is 0 for a user-included reagent that did not appear in the CombiGlide docking output. The structure in which each reagent first appeared in the docking output is given, as well as its index in the .bld file. These values pertain to this modification of the reagent.

For the `boltz` strategy, the Boltzmann population and the computed `Delta_G` of each reagent is given, relative to the most frequent reagent (in the Boltzmann sense) at its position.

For each reagent there is a 1 in the `b_first` column if it is the first modification of that reagent to appear in this position of the output, and a 0 otherwise. Thus, by counting the 1's, you can count the reagents, rather than the modifications.

A 1 is placed in the `b_include` column to indicate that the reagent was specified on the user's list of reagents always to include at that position. Finally, the reagent name is reported last on each line.

Following the `number_selected:` lines of selected reagents, a few runners-up are listed, as described earlier.

Creating Custom Functional Groups and Custom Minimal Capping Groups

This appendix describes the procedures for adding custom functional groups or modifying existing functional groups.

The definitions of the functional groups are stored in two files in the CombiGlide distribution, `reagentprep.ini`, which is a plain text file, and `reagentprep.mae`, which is a structure file in Maestro format. These files may be found in the directory `$SCHRODINGER/combiglide-v`*version*`/data`. The functional group definitions include the following:

- **A long name.** This name is used to identify the functional group, and is used for its title. The long name is displayed in a tooltip when you pause the pointer over the reagent button in the Reagent Preparation panel. The name can be chosen any way you wish, but must not contain blanks. The long names for the supplied functional groups are constructed by adding the atom that is kept and the atom that is lost to the functional group name, separated by underscores, for example `Acid_Chloride_C_C`.
- **A short name.** This is the name that is displayed on the reagent button in the Reagent Preparation panel. The short names for the supplied functional groups are constructed by adding the atom that is kept and the atom that is lost to the shortened functional group name, separated by underscores, for example `Acid_Cl_C_C`. Because the reagent button displays the short name for custom functional groups, it is advisable to ensure that you can identify both the functional group and the bond that is broken in the name. Adopting the convention used for the supplied functional groups is one way to do this.
- **One or more SMARTS patterns that define the group.** The SMARTS patterns must be contiguous, without any spaces. At the top of `reagentprep.ini` is a description of the syntax with which you can use multiple SMARTS patterns to define the functional group.
- **A grow bond definition**. The grow bond defines the bond to be broken when the side chain from the reagent is added to the core. The bond is defined in terms of two atoms, the atom that is kept, and the atom that is deleted. This bond must be a single bond and must not be internal to a ring.
- **A structure that defines the minimal capping group.** This is the group that is added at the attachment position to define the minimally capped core, and in the construction of the singly substituted core molecules. The structure should consist of the desired functional group with the minimal capping group attached. For example, if you wanted to define an acid chloride with a methyl group as a minimal capping group, you would choose acetyl chloride as the structure. The structure should be 3D and minimized.

To add a new functional group, you must edit the file `reagentprep.ini` and add one or more lines that include the long name, the SMARTS pattern, and the grow bond definition, and you must add a structure to the file `reagentprep.mae`, with the long name, the short name, and the grow bond defined.

To change the capping group for an existing functional group, you must modify the structure in the file `reagentprep.mae`.

You can store the modified files in the directory `$HOME/.schrodinger/combiglide` or the directory from which you submit your reagent preparation job. When a reagent preparation job is run, CombiGlide looks first in the job submission directory for these files, then in `$HOME/.schrodinger/combiglide`, then in `$SCHRODINGER/combiglide-v`*version*`/data`. If you want to customize these files just for a particular job, you should store them in the job submission directory. If you want to use the custom functional groups for all your jobs, you should store the files in `$HOME/.schrodinger/combiglide`.

To add a functional group definition to reagentprep.ini:

1. Copy the file `reagentprep.ini` to the desired location.
2. Open `reagentprep.ini` in a text editor.
3. Insert a line into the file with the following format, followed by a blank line:

   `AddGN` *longname* *SMARTSpattern* *kept-atom*`:rpc1` *lost-atom*`:rpc2`

   An example for the acid chloride functional group is:

   ```
   AddGN Acid_Chloride_C_Cl [CX3,c](=O)(Cl)[#6] 1:rpc1 3:rpc2
   ```

   The variables that you need to replace have the following meanings:

longname	The long name of the functional group (see above).
SMARTSpattern	SMARTS pattern that defines the functional group.
kept-atom	The index of the atom in the grow bond that is kept when the bond is broken and the side chain is added to the core. The index is the index in the SMARTS string.
lost-atom	The index of the atom in the grow bond that is lost when the bond is broken and the side chain is added to the core. The index is the index in the SMARTS string.

   `rpc1` and `rpc2` are the *grow names* for the specified atoms, which are used to identify the kept atom and the lost atom when the side chain is "grown" onto the core.

4. Save and close the file.

To add a structure for the functional group to reagentprep.mae:

1. Copy the file `reagentprep.mae` to the desired location.
2. Start Maestro and open the Project Table panel (click the toolbar button shown below).

3. Add the desired structure to the Project Table.

   You can import the structure using the Import panel, or you can build it, or duplicate one of the structures in the Project Table and modify it. It might be useful to import the structures from `reagentprep.mae` to duplicate and modify.

   The structure must consist of the functional group and the minimal capping group only. The structure must be a 3D minimized structure. If it is not, you can convert and minimize it with LigPrep, or choose Clean Up Geometry from the Edit menu in the main window to minimize it with the UFF minimizer.

   If you build the structure, you must add it as an entry to the Project Table, by clicking the Create entry from Workspace toolbar button.

   The entry name does not matter, but the title must be the long name of the functional group. If you make the entry name the long name, the title is automatically copied from the long name and you do not need to change it in Step 5. You can also add the structure to the entry group that was imported, but it is not necessary to do so.
4. Enter the long name and the short name in the appropriate columns of the Project Table for the new structure.
5. Enter the long name as the title of the new structure.

   For both of these steps, you should ensure that the long name is the same as in the file `reagentprep.ini`.

Now you define the grow bond.

6. Ensure that the structure is displayed in the Workspace.
7. From the Display menu in the main window, choose Atom Labels.

   The Atom Labels panel opens.
8. In the Composition folder, select Grow Name and deselect all other labels.

9. Click All in the Label Atoms section.

   The atoms are now labeled with their grow names, if they have one.

10. Open the Build panel.

11. In the Atom Properties folder, choose Grow Name from the Property option menu.
12. Ensure that Atoms is chosen from the Pick option menu in the Apply grow name section.
13. Type `rpc1` into the Grow Name text box, and click on the atom in the Workspace that is to be kept when the bond is broken.
14. Type `rpc2` into the Grow Name text box, and click on the atom in the Workspace that is to be lost when the bond is broken.

    If you make a mistake with any of these assignments, just type the correct name in and pick the atom again. These two atoms must correspond to the atoms in the SMARTS pattern whose indices you entered in `reagentprep.ini`.

    To delete a grow name, enter two successive quotation marks "" (a null string) into the Grow Name text box.

Finally, you add the new structure to the file.

15. Choose Export Structures from the Project menu in the main window.

    The Export panel opens.

16. Navigate to your copy of `reagentprep.mae` and select it.
17. Under the File text box, select Append.
18. Under Structure sources to be exported, select Workspace.
19. Ensure that Export all entries to a single file is selected from the Files option menu.
20. Click Export.

    The entry displayed in the Workspace, which is your new structure, is appended to `reagentprep.mae`

If you want to add multiple functional groups, you can repeat the relevant steps above for other structures. When you come to adding the new structures, select the entries in the Project Table, and select Selected entries instead of Workspace under Structure sources to be exported in the Export panel (Step 18).

To modify a minimal capping group:

1. Copy the file `reagentprep.mae` to the desired location.
2. Start Maestro and open the Project Table panel (click the toolbar button shown below).

   If you start Maestro in the directory to which you copied `reagentprep.mae`, no navigation is required when you import or export the file.
3. Import `reagentprep.mae`.

   
4. Include the structure for the reagent you want to modify in the Workspace.
5. Modify the structure.

   You can use the Build panel to modify or add to the structure.
6. Minimize the structure.

   The preferred treatment is to use MacroModel with the OPLS_2005 force field. Minimization is important because the docking process does not change the bond lengths and bond angles in the single-position docking step.
7. Select all the structures that were imported.

   This should be all the structures in the Project Table, if you started with a new Maestro session.
8. Choose Export Structures from the Project menu in the main window.

   The Export panel opens.
9. Navigate to your copy of `reagentprep.mae` and select it.
10. Under the File text box, deselect Append.
11. Under Structure sources to be exported, select Selected entries.
12. Ensure that Export all entries to a single file is selected from the Files option menu.
13. Click Export.

    A dialog box is displayed to confirm the overwriting of `reagentprep.mae`
14. Click Yes.

The file `reagentprep.mae` is overwritten with the modified structure set.

To replace the existing structure for a capping group with another structure:

1. Follow the instructions for adding a structure above, from Step 1 through Step 14.

   When you enter the title, long name, and short name, use the values for the entry containing the structure you want to replace.

2. Delete the entry that you want to replace.

   The new entry has all the required information in it, and the order of the entries does not matter. If you want to move the new entry to the position that the deleted entry was in, select the entry and drag it to the desired location, or choose Move to Row from the Entry menu.

3. Select all the structures that were imported and the new structure.

   If you started with a new Maestro session, this should be all structures in the Project Table.

4. Follow the instructions for modifying a minimal capping group, from Step 8 through Step 14.

Command-Line Tools

The CombiGlide distribution includes a number of utilities, which are installed into `$SCHRODINGER/utilities`, and are described below.

B.1 cg_add_chem_features

This utility adds chemical features to CombiGlide reagent files that were originally prepared without them, so that chemical feature analysis can be performed on an already-completed CombiGlide run. Starting with Schrodinger Suite 2007, chemical features are added to all newly prepared reagent files automatically. Reagent files are identified by a `.bld` suffix. Along with this file there are also an `.sqlite` and a `_vpost-stats.txt` file, with the same base name. These three files are required by CombiGlide when adding a reagent.

The syntax is as follows:

`cg_add_chem_features [-h]` *reagent-file* `[`*reagent-file2* `... ]`

The reagent file name can be given with or without the `.bld` suffix. This file must include a block at the top containing an `s_rp_functionalgroup` property, which will always be the case if the file was created by means of the CombiGlide reagent preparation procedure.

The utility writes a new `.bld`, `.sqlite` and `_vpost-stats.txt` file for each input file, with the same names as the input files. The new files are always created in the current directory. If you want to avoid overwriting the old files, change to the directory in which you want to write the new files, and specify the input files using relative or absolute paths.

If the utility is run from a directory containing the old files, an attempt will first be made to back up the old files by appending `-bak` to their names, so *myfile*`.bld` is copied to *myfile*`.bld-bak`, and likewise for the `.sqlite` and `_vpost-stats.txt` files. If both the *myfile*`.bld` and the *myfile*`.bld-bak` files exist, the utility assumes that a previous run failed, and uses the `-bak` files as input and overwrites the original files.

To add chemical features to reagent files used in earlier runs of CombiGlide, you need to run the program on the copies of the `.bld` files that were stored when the run was done. If the run was done from a Maestro project *project*`.prj`, the files are in directories named *project*`.prj/.mmproj-admin/vcs.`*N* and *project*`.prj/.mmproj-admin/vcs.`*N*`/combiglide`, where *N* is an integer. You should process both sets of `.bld` files.

Finally, some Schrödinger programs require `.mae` suffixes. To accommodate these programs, a *myfile*`.mae` symbolic link is temporarily made to the old reagent file *myfile*`.bld`, and the link is removed when it is no longer needed. If *myfile*`.mae` already exists when the utility is executed, the utility fails.

B.2 chem_features

This utility lists the chemical feature types present in the input structures. Several definitions of feature types can be used: the CombiGlide definition, the Phase definition, or a user-specified definition. See Section 11.1.1 on page 86 for a description of the standard feature definitions used by CombiGlide.

The command syntax is as follows:

`chem_features [-h] [-v` *level*`] [-r] [-R` *recapped-file*`] [-p]`
`[-f {P|C|`*filename*`}] [-q` *logP-cutoff*`|-Q] [-o` *output-file*`]` *input-file*

In the absence of `-h` or `-v` options or errors, there is no text output. The options are described in detail in Table B.1.

Table B.1. Options for the chem_features utility.

Option	Description
General options	
`-h`	Display usage message.
`-n` *number*	Process the first *number* structures from the input file only.
`-v` *level*	Verbosity level 0 Silent (unless -h is specified); 1 Feature and property information only 2 Add information about operation of the program. The verbose output, if any, is written to standard output.
`-f {C|P|`*filename*`}`	Specify which initialization file to use. `C` Use the CombiGlide chemical features file `chem_feature.ini`. This is the default. P Use the standard Phase feature definition file, `pharma_feature.ini`. *filename* Use the file specified. Anything other than `C` or `P` is interpreted as a file name. The location used for the feature definition file for both CombiGlide and Phase is the first one in the following list in which the file is found: • The local directory • `$HOME/.schrodinger/mmshare` • `$SCHRODINGER/mmshare-v`*version*`/data`

Table B.1. Options for the chem_features utility.

Option	Description
`-q` *cutoff*	Specify the cutoff value for the QikProp property QPlogPo/w to use when determining whether a structure is hydrophobic.The default is 2.5 if `-f C` is specified or `-f` is not used. Otherwise, no criterion is used.
`-Q`	Do not use QPlogPo/w cutoff values in determining whether a structure has hydrophobic features. This is the default unless `-f C` is specified or implied.
`-o` *filename*	Write an output Maestro file that contains the same structures as the input file. If `-p` is specified, the computed properties are added to each structure.
CombiGlide-specific options	
`-r`	Replace CombiGlide reactive functional groups with $–CH_3$ before identifying features.
`-R` *filename*	Write the capped structures generated by `-r` to *filename*. This option implicitly sets `-r`, so there is no need to use `-r` with `-R`.
`-p`	Compute structure-level Maestro properties to indicate presence of features. If there is a capping atom, the "maximum path" for each pharmacophore type is also computed. Properties are added to the following files: • The structure output file, if `-o` is specified. • The recapped output file if `-R` is specified. • Standard output if `-v` *N* is specified with *N*>0.

Path lengths are computed as follows:

- For each atom in each pharmacophore, the minimum number of bonds between each atom in the pharmacophore and the capping atom is computed.
- The maximum of these values for the feature is the maximum path for that feature.
- The maximum of these values for all features of the same feature type (e.g., hydrophobes) is the value reported for each feature type.

If paths are computed then, for each feature type, properties `i_chemfeatures_`*type*`_atom` and `i_chemfeatures_`*type*`_dist` are computed. The `_dist` property contains the maximum path for that feature type and the `_atom` property contains an atom number exhibiting that distance. The possible values of *type* are `hydrophobe`, `donor`, `acceptor`, `positive`, `negative`, and `aromatic`.

If a feature of type *type* is not present, then the `_atom` and `_dist` properties for that type are given values of 0.

If there is no capping atom but there is an atom with growname `rpc2`, it is used to compute paths.

If the type is present but there is no capping atom, the `_atom` feature is given a value of -1 and the `_dist` feature is given a value of 0. Thus, if `-p` is specified, the `_atom` property for any feature type is zero if the feature is absent and non-zero if it is present.

Examples:

List default chemical features for the structures in `input.mae`:

```
$SCHRODINGER/utilities/chem_features -v 1 input.mae
```

List Phase pharmacophoric features for the structures in `input.mae`:

```
$SCHRODINGER/utilities/chem_features -v 1 -f P input.mae
```

Normal CombiGlide use, during reagent preparation, replacing the reactive side chain with $–CH_3$ and producing file `output.bld` containing the properties representing the properties in `input.bld`:

```
chem_features -r -p -q 2.5 -o output.bld input.bld
```

Verbose CombiGlide use, producing a file containing the capped structures as well as verbose output concerning operation of the program):

```
chem_features -v 2 -R capped.bld -p -q 2.5 -o output.bld input.bld
```

B.3 cg_combine_bld

This utility combines a set of reagent (`.bld`) files that were created by the Reagent Preparation procedure so that the combined file can be used as a reagent file at a single position in combinatorial enumeration or screening. Structures whose R-groups are identical to ones in previous structures processed are discarded.

The utility uses the functional group definition found in the first input file specified as the functional group definition in the output file. The functional group specification has the effect of defining the minimal capping group in the Combinatorial Screening workflow. However, since minimal capping groups are not used by default, the order of specification of input files does not matter unless minimal capping groups are explicitly used.

The command syntax is as follows:

```
cg_combine_bld [-f] [-h] [-v] [-doc] input-stem [ input-stem... ]
    output-stem
```

The options are described in Table B.2. Each *input-stem* is the file name stem of a `.bld` file (file name minus the `.bld` extension). The output consists of the *output-stem*`.bld`, *output-*

*stem*_vpost-stats.txt, and *output-stem*.sqlite files. The last two files allow the output .bld file to be used as a reagent file in combinatorial screening.

Table B.2. Options for the cg_combine_bld command.

Option	Description
-v[ersion]	Show the program version and exit.
-h[elp]	Show the usage message and exit.
-f	Force overwrite of pre-existing output files.
-doc	Print detailed documentation.

Getting Help

Schrödinger software is distributed with documentation in PDF format. If the documentation is not installed in `$SCHRODINGER/docs` on a computer that you have access to, you should install it or ask your system administrator to install it.

For help installing and setting up licenses for Schrödinger software and installing documentation, see the *Installation Guide*. For information on running jobs, see the *Job Control Guide*.

Maestro has automatic, context-sensitive help (Auto-Help and Balloon Help, or tooltips), and an online help system. To get help, follow the steps below.

- Check the Auto-Help text box, which is located at the foot of the main window. If help is available for the task you are performing, it is automatically displayed there. Auto-Help contains a single line of information. For more detailed information, use the online help.
- If you want information about a GUI element, such as a button or option, there may be Balloon Help for the item. Pause the cursor over the element. If the Balloon Help does not appear, check that Show Balloon Help is selected in the Maestro menu of the main window. If there is Balloon Help for the element, it appears within a few seconds.
- For information about a panel or the tab that is displayed in a panel, click the Help button in the panel, or press F1. The help topic is displayed in your browser.
- For other information in the online help, open the default help topic by choosing Online Help from the Help menu on the main menu bar or by pressing CTRL+H. This topic is displayed in your browser. You can navigate to topics in the navigation bar.

The Help menu also provides access to the manuals (including a full text search), the FAQ pages, the New Features pages, and several other topics.

If you do not find the information you need in the Maestro help system, check the following sources:

- *Maestro User Manual*, for detailed information on using Maestro
- *Maestro Command Reference Manual*, for information on Maestro commands
- *Maestro Overview*, for an overview of the main features of Maestro
- *Maestro Tutorial*, for a tutorial introduction to basic Maestro features
- *CombiGlide Quick Start Guide*, for a tutorial introduction to CombiGlide
- *Glide User Manual*, for detailed information on using Glide
- *Protein Preparation Guide*, for information on protein preparation

- CombiGlide Frequently Asked Questions pages, at https://www.schrodinger.com/CombiGlide_FAQ.html
- Known Issues pages, available on the Support Center.

The manuals are also available in PDF format from the Schrödinger Support Center. Local copies of the FAQs and Known Issues pages can be viewed by opening the file `Suite_2009_Index.html`, which is in the `docs` directory of the software installation, and following the links to the relevant index pages.

Information on available scripts can be found on the Script Center. Information on available software updates can be obtained by choosing Check for Updates from the Maestro menu.

If you have questions that are not answered from any of the above sources, contact Schrödinger using the information below.

E-mail: help@schrodinger.com
USPS: Schrödinger, 101 SW Main Street, Suite 1300, Portland, OR 97204
Phone: (503) 299-1150
Fax: (503) 299-4532
WWW: http://www.schrodinger.com
FTP: ftp://ftp.schrodinger.com

Generally, e-mail correspondence is best because you can send machine output, if necessary. When sending e-mail messages, please include the following information:

- All relevant user input and machine output
- CombiGlide purchaser (company, research institution, or individual)
- Primary CombiGlide user
- Computer platform type
- Operating system with version number
- CombiGlide version number
- Maestro version number
- mmshare version number

On UNIX you can obtain the machine and system information listed above by entering the following command at a shell prompt:

```
$SCHRODINGER/utilities/postmortem
```

This command generates a file named *username*`-`*host*`-schrodinger.tar.gz`, which you should send to help@schrodinger.com. If you have a job that failed, enter the following command:

```
$SCHRODINGER/utilities/postmortem jobid
```

where *jobid* is the job ID of the failed job, which you can find in the Monitor panel. This command archives job information as well as the machine and system information, and includes input and output files (but not structure files). If you have sensitive data in the job launch directory, you should move those files to another location first. The archive is named *jobid*`-archive.tar.gz`, and should be sent to help@schrodinger.com instead.

If Maestro fails, an error report that contains the relevant information is written to the current working directory. The report is named `maestro_error.txt`, and should be sent to help@schrodinger.com. A message giving the location of this file is written to the terminal window.

More information on the `postmortem` command can be found in Appendix A of the *Job Control Guide*.

On Windows, machine and system information is stored on your desktop in the file `schrodinger_machid.txt`. If you have installed software versions for more than one release, there will be multiple copies of this file, named `schrodinger_machid-`*N*`.txt`, where *N* is a number. In this case you should check that you send the correct version of the file (which will usually be the latest version).

If Maestro fails to start, send email to help@schrodinger.com describing the circumstances, and attach the file `maestro_error.txt`. If Maestro fails after startup, attach this file and the file `maestro.EXE.dmp`. These files can be found in the following directory:

```
%USERPROFILE%\Local Settings\Application Data\Schrodinger\appcrash
```

Glossary

Attachment—Combination of an attachment position and a reagent file.

Attachment position—Position on a core to which side chains are attached (synonymous with 'site of diversity').

Chemical feature—One of six standard types of functional group: hydrogen bond acceptor (A), hydrogen bond donor (D), hydrophobe (H), negatively charged center (N), positively charged center (P), or aromatic (R). The CombiGlide definition of these feature types differs somewhat from that of standard pharmacophore analysis.

Combinatorial library—A set of compounds prepared by combinatorial synthesis.

Combinatorial synthesis—The synthesis of sets of compounds utilizing the same reaction sequence where all possible combinations of the reagents are used.

Compound—In CombiGlide, a compound is explicitly defined by a combination of reagents that are attached to a core to form a product. A compound may encompass several structures. If a reagent exists in more than one form (different tautomers, conformers, stereoisomers, protonation state), the compound may also encompass more than one variant.

Core—The structural element of a combinatorial library that is constant throughout the library. It is the part of the core-containing molecule that remains when the side chains are removed.

Enumerate—To generate structures of all possible members of a combinatorial library.

Library selection—The results of filtering and selection that generates a list of reagents at each position for a focused library, including information on the filtering and selection strategy.

Fully substituted core—A core with side chains attached at all attachment positions.

Library definition—A collection of the core-containing structure and the reagent files that defines a combinatorial library. Can be used as input to library enumeration or combinatorial screening.

Minimal capping group—The smallest representative side chain for a particular functional group.

Minimally capped core—A core with the appropriate minimal capping group at each attachment position.

Protocore—A molecule that could be used to replace the core molecule, and that has hydrogen atoms where the side chains should be attached.

Reagent—The chemical reagent used in the combinatorial synthesis; a structure or set of structures with the same chemical composition and heavy-atom connectivity that is used to add a side chain to the core. A reagent may encompass more than one structure if it exists in more than one form (different tautomers, conformers, stereoisomers, protonation state).

Reagent variant—A single reagent structure, defined by a particular tautomer, conformer, stereoisomer, and protonation state of the reagent.

Side chains—The structural elements of a combinatorial library that are varied within the library. The side chain is the part of a reagent that remains after removing the reactive functional group that is discarded when it is bound to the core.

Structure—A molecule with a particular geometry and stereochemistry, in a single tautomeric form, ionization state, and conformation.

User-capped core—The molecule that was used to define the core. The capping groups are the side chains from this molecule.

Type—The list of chemical features that can be found in a side chain.

Index

M

O

P

Q

R

S

U

120 West 45th Street, 29th Floor
New York, NY 10036

101 SW Main Street, Suite 1300
Portland, OR 97204

8910 University Center Lane, Suite 270
San Diego, CA 92122

Zeppelinstraße 13
81669 München, Germany

Dynamostraße 13
68165 Mannheim, Germany

Quatro House, Frimley Road
Camberley GU16 7ER, United Kingdom

SCHRÖDINGER®